



**Titre:** Étude préliminaire de l'utilisation de la réalité virtuelle pour  
Title: l'analyse de tolérance

**Auteur:** Rémy Dionne  
Author:

**Date:** 2007

**Type:** Mémoire ou thèse / Dissertation or Thesis

**Référence:** Dionne, R. (2007). Étude préliminaire de l'utilisation de la réalité virtuelle pour  
Citation: l'analyse de tolérance [Mémoire de maîtrise, École Polytechnique de Montréal].  
PolyPublie. <https://publications.polymtl.ca/8080/>

 **Document en libre accès dans PolyPublie**  
Open Access document in PolyPublie

**URL de PolyPublie:** <https://publications.polymtl.ca/8080/>  
PolyPublie URL:

**Directeurs de  
recherche:**  
Advisors:

**Programme:** Non spécifié  
Program:

UNIVERSITÉ DE MONTRÉAL

ÉTUDE PRÉLIMINAIRE DE L'UTILISATION DE LA RÉALITÉ VIRTUELLE POUR  
L'ANALYSE DE TOLÉRANCE

RÉMY DIONNE  
DÉPARTEMENT DE GÉNIE MÉCANIQUE  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION  
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES  
(GÉNIE MÉCANIQUE)  
AOÛT 2007

© Rémy Dionne, 2007.



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file    Votre référence*

*ISBN: 978-0-494-36908-1*

*Our file    Notre référence*

*ISBN: 978-0-494-36908-1*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

UNIVERSITÉ DE MONTRÉAL  
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

ÉTUDE PRÉLIMINAIRE DE L'UTILISATION DE LA RÉALITÉ VIRTUELLE POUR  
L'ANALYSE DE TOLÉRANCE

présenté par: DIONNE Rémy

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment acceptée par le jury d'examen constitué de:

M. MAYER René, Ph.D., président

M. BARON Luc, Ph.D., membre et directeur de recherche

M. MASCLE Christian, Doct.ès.sc., membre et codirecteur de recherche

M. DESROCHERS Alain, Dr.ing., membre

## REMERCIEMENTS

J'aimerais tout d'abord remercier mes directeur et co-directeur de m'avoir épaulé, fait confiance et soutenu tout au long de cette recherche dans un domaine qui rassemble à la fois les concepts mathématiques de la robotique, la fabrication et l'assemblage. Ce fut pour moi un défi et une expérience extraordinaire.

Merci Luc Baron de votre support et de m'avoir donné la chance d'évoluer au sein du laboratoire CAE en modélisation et simulation. Les ressources de qualité mises à ma disposition ont été d'une grande aide et furent grandement appréciées. Ces années de formation furent des plus enrichissantes autant sur le plan personnel que professionnel.

Merci Christian Mascle de m'avoir accompagné tout au long de la réalisation de ce projet. Vos précieux conseils et votre expertise professionnelle ont souvent permis de me remettre sur le droit chemin. Ce fut un privilège de cheminer sous votre direction.

Dans un deuxième temps, j'aimerais également remercier tous les étudiants du laboratoire ayant été impliqués de près ou de loin pour leurs précieux conseils. Plus particulièrement, merci à Sofiane Achiche, Liguó Huo et Xiaoyu Wang.

Finalement, j'aimerais remercier ma famille et mes amis, dont plus particulièrement Annie Levasseur, qui m'ont soutenu tout au cours de ce long processus de formation. Leur présence et leur écoute attentive autant dans les moments de réjouissance que de tourment furent très appréciées.

## RÉSUMÉ

Dans le contexte actuel de mondialisation, les compagnies ont de plus en plus recours à la sous-traitance. Lors de la conception de produit, il est primordial d'assurer que les composantes fabriquées dans des endroits différents puissent s'assembler. Le tolérancement géométrique et dimensionnel des pièces est une information qui prend une valeur légale dans certains cas et qui permet d'assurer la conformité des pièces. Cependant, le coût de fabrication est proportionnel à la précision requise. Une grande précision d'usinage permet de minimiser les rejets de pièces et les problèmes d'assemblages. D'un autre côté, les coûts de fabrication afférents sont plus élevés. Des tolérances trop grandes diminueront les coûts de fabrication, mais augmenteront les problèmes de qualité. L'optimisation des tolérances est possible grâce à des outils d'analyse.

L'objectif visé par cette recherche est de résoudre le problème d'analyse des tolérances qui se conceptualise difficilement. L'analyse des tolérances géométriques et dimensionnelles se divise en deux étapes principales : (1) l'interprétation des tolérances pour obtenir les boucles cinématiques d'une fonction de contrôle et capter la signification des tolérances et (2) une méthode de résolution afin de chiffrer la grandeur des jeux ou serrages.

Le premier objectif de cette recherche consistait à définir une nouvelle méthode de résolution graphique et en démontrer le concept dans un environnement de réalité virtuelle. Le deuxième objectif était d'assurer que toutes les tolérances soient considérées dans l'analyse.

Le premier objectif est comblé par la réalisation d'une application prototype qui utilise une méthode graphique 3D pour l'analyse des tolérances. La faisabilité de l'approche est démontrée dans un environnement de réalité virtuelle ce qui permet de mieux saisir la signification des jeux. Des améliorations ou solutions sont proposées pour remédier aux

problèmes rencontrés lors du développement. En deuxième lieu, une librairie C++ pour interpréter les tolérances a été développée afin de rencontrer le deuxième objectif. Cet outil permet de générer les boucles cinématiques d'une fonction de contrôle et d'identifier les tolérances qui affectent chaque élément topologique.

## ABSTRACT

In today's trend towards globalization, companies are subcontracting an increasing amount of their workload. It is of primary importance to ensure that the components manufactured in different places will assemble. The geometrical and dimensional tolerances on the detail drawing indicate the quality standards of a part. However, the cost of manufacturing is proportional to the requested precision level. A high degree of accuracy for machining makes it possible to minimize the rejections of parts that would result from non-assemblability and other quality problems, but the manufacturing costs are higher. Large tolerances can reduce the manufacturing costs, but will decrease the overall quality. Tools to carry out the analysis of the tolerances make it possible to optimize the tolerances.

The aim of this research is to solve the problem of tolerance analysis which is difficult to conceptualize. The analysis of the geometrical and dimensional tolerances is divided into two main tasks: (1) the interpretation of the tolerances to capture the meaning of the tolerances and obtain the kinematics loops of the control function and (2) a method of resolution in order to quantify the size of the gaps or interference.

The first objective of this research was to define a new graphical method of resolution and demonstrate its concept in a virtual reality environment. The second objective was to ensure that any type of tolerance could be included in the analysis.

The first objective is met through the development of a prototype application which uses a 3D graphical method to perform tolerance analysis in a virtual reality environment. This demonstrates the feasibility of the approach and recommendations are proposed to enhance the application. The use of virtual reality is justified because it makes it possible to better seize the impact of the tolerances. For the second objective, a C++ library has been developed to do the interpretation of the tolerances. This tool makes it possible to



generate the kinematic loops of a control function and identify the tolerances that affect any topological element in the loop.

## TABLE DES MATIÈRES

REMERCIEMENTS.....	IV
RÉSUMÉ.....	V
ABSTRACT.....	VII
TABLE DES MATIÈRES.....	IX
LISTE DES TABLEAUX.....	XIII
LISTE DES FIGURES.....	XIV
LISTE DES SIGLES ET ABRÉVIATIONS.....	XVIII
LISTE DES ANNEXES.....	XIX
AVANT-PROPOS.....	XX
INTRODUCTION.....	1
 CHAPITRE 1 : REVUE CRITIQUE DE LA LITTÉRATURE.....	 4
1.1 Logiciels commerciaux pour l'analyse des tolérances.....	4
1.1.1 Catia V5.....	4
1.1.2 TASYSSWorks.....	5
1.1.3 Cetol Six Sigma.....	6
1.1.4 3DCS.....	7
1.1.5 VisVSA.....	7
1.2 État de la recherche.....	7
1.2.1 Analyse 1D.....	9
1.2.2 Analyse 2D.....	10
1.2.3 Analyse 3D.....	11
1.2.4 Le graphe de pièce.....	17
 CHAPITRE 2: APPLICATION EN RÉALITE VIRTUELLE.....	 25
2.1 Démarche de l'ensemble du travail de recherche.....	25
2.2 Présentation des outils informatiques.....	27
2.2.1 L'environnement de réalité virtuelle.....	27

2.2.2 Description des bibliothèques utilisées pour l'application en réalité virtuelle .....	28
2.2.3 Les menus .....	29
2.2.4 Structure du logiciel.....	29
2.2.5 Structure de données : aperçu du graphe de scène .....	31
2.3 L'interface utilisateur.....	31
2.4 Analyse des tolérances.....	36
2.5 Situation des articles dans le mémoire.....	41

### CHAPITRE 3: ARTICLE I: METHODOLOGY FOR THE INTERPRETATION OF GEOMETRICAL AND DIMENSIONAL TOLERANCES IN AN

ASSEMBLY .....	42
3.1 Abstract.....	43
3.2 Key words.....	43
3.3 Introduction.....	43
3.4 State of the Art.....	45
3.4.1 Commercial Software .....	45
3.4.2 Review of the Literature .....	47
3.5 Basic Concepts.....	50
3.5.1 Definition of the G&D Tolerances .....	50
3.5.2 Precedence of the Tolerances .....	52
3.5.3 Datums .....	53
3.5.4 Worst cases .....	54
3.5.5 Rules for Constructing the Reference Frames .....	55
3.6 Proposed Approach.....	55
3.6.1 Creation of a Part Graph.....	56
3.6.2 Creation of an Assembly Graph .....	60
3.7 Implementation .....	62
3.7.1 Resolution by the Traditional Uniaxial Method .....	64
3.7.2 Resolution by the Suggested Approach.....	68

3.8 Comparison of the Two Methods .....	77
3.9 Prototype Application for 3D Tolerance Analysis in a VR Environment .....	79
3.10 Conclusion .....	80

CHAPITRE 4: ARTICLE II: A C++ LIBRARY FOR THE AUTOMATIC INTERPRETATION OF GEOMETRICAL AND DIMENSIONAL TOLERANCES.....	81
4.1 Abstract.....	82
4.2 Key words.....	82
4.3 Introduction.....	82
4.4 State of the Art.....	84
4.5 Implementation of the C++ Library.....	86
4.5.1 Tolerances.....	87
4.5.2 Part Class .....	89
4.5.3 Assembly Class.....	90
4.6 Interpretation of the Tolerances.....	92
4.6.1 Identification of the Paths.....	92
4.6.2 Retrieving the Tolerance Dependencies of a Topological Element.....	96
4.6.3 Relative Positioning Between Two Toleranced Topological Elements .....	99
4.6.4 Method of Resolution .....	99
4.7 Conclusion.....	102
CHAPITRE 5 : DISCUSSION GÉNÉRALE .....	103
5.1 Difficultés rencontrées avec la génération des volumes d'existence .....	103
5.2 Résultats obtenus lors de l'interprétation des tolérances.....	107

CONCLUSION ET RECOMMANDATIONS.....	114
RÉFÉRENCES .....	117
ANNEXES .....	122

**LISTE DES TABLEAUX**

Table 3.1.	The transformations for the three paths of the block on rail assembly analysis.....	71
Table 3.2.	The transformations of Path 2.....	72

## LISTE DES FIGURES

Figure 1.1.	Aperçu de l'atelier <i>Functional Tolerancing and Annotations</i> (FT&A) de Catia V5 [1].....	4
Figure 1.2.	Relation entre les éléments du modèle paramétrique abstrait [2] : a) contrainte plan-plan parallèle ; b) contrainte plan-ligne parallèle ; c) contrainte plan-point ; d) contrainte ligne-ligne parallèle ; e) contrainte ligne-ligne co-planaire ; f) contrainte ligne-point ; g) contrainte point-point ; h) contrainte plan-plan non-parallèle ; i) contrainte plan-ligne non-parallèle ; j) contrainte ligne-ligne non-planaire ; k) contrainte de forme ; l) contrainte de cylindricité ; m) contrainte de circularité.....	13
Figure 1.3.	Dessin de détail partiel d'une pièce [17].....	18
Figure 1.4.	Fragment de VGraph [18].....	18
Figure 1.5.	Pièce tolérancée [18].....	20
Figure 1.6.	Identification de la topologie [18].....	20
Figure 1.7.	Graphe de pièce (légèrement modifié) [18].....	20
Figure 1.8.	Pièce avec 4 systèmes de référence inter-reliés [20] .....	22
Figure 1.9.	Graphe de la pièce à la figure 1.8 [20].....	22
Figure 1.10.	Exemple d'un assemblage non tolérancé [21] .....	24
Figure 1.11.	Le réseau de contraintes et le graphe des tolérances générées [21].....	24
Figure 2.1.	Description de la CAVE .....	27
Figure 2.2.	Structure du logiciel.....	29
Figure 2.3.	Description du graphe de scène utilisé.....	30
Figure 2.4.	Hierarchie des menus.....	32
Figure 2.5.	Aperçu du menu principal en mode simulation .....	32
Figure 2.6.	Aperçu du sous-menu de fichier .....	33
Figure 2.7.	Interface usager pour la sélection d'un modèle à charger.....	33
Figure 2.8.	Aperçu du sous-menu de surfaces.....	34
Figure 2.9.	Sélection d'une face cylindrique en utilisant la boîte de sélection 3D .....	34

Figure 2.10.	Résultat de la sélection d'une face cylindrique.....	35
Figure 2.11.	Aperçu du sous-menu des tolérances géométriques .....	35
Figure 2.12.	Aperçu du sous-menu des liaisons d'assemblages.....	35
Figure 2.13.	Volume d'existence d'une zone de tolérance .....	35
Figure 2.14.	Aperçu du sous-menu d'analyse des tolérances.....	36
Figure 2.15.	Pièce cylindrique et sa zone de tolérance .....	37
Figure 2.16.	Manipulation de la zone de tolérance .....	37
Figure 2.17.	Volume d'existence d'une zone de tolérance .....	37
Figure 2.18.	Procédure de génération du volume d'existence .....	39
Figure 2.19.	Les quatre directions de rotation.....	40
Figure 2.20.	Angle de rotation maximal selon l'axe X3 .....	40
Figure 3.1.	Examples of dimensional tolerances.....	50
Figure 3.2.	Summary of the geometrical tolerances (ASME Y14.5M-1994 [29]) .....	51
Figure 3.3.	Controlling the positioning, orientation and form of a hole's axis .....	53
Figure 3.4.	Representation of the three tolerance zones of the hole's axis .....	53
Figure 3.5.	Types of tolerance zones and the degrees of freedom they constrain.....	54
Figure 3.6.	Geometrical and dimensional annotations on a part .....	58
Figure 3.7.	Identification of the topological elements of the part .....	58
Figure 3.8.	Example of a part graph.....	59
Figure 3.9.	The dependency tree for surfaces S7 and S4 .....	60
Figure 3.10.	Types of assembly constraints, their symbol and affected DOFs.....	61
Figure 3.11.	Example of an assembly graph .....	62
Figure 3.12.	The block on rail assembly .....	63
Figure 3.13.	Detail drawing of the block.....	64
Figure 3.14.	Detail drawing of the rail .....	64
Figure 3.15.	Minimum gap configuration for the analysis.....	66
Figure 3.16.	Maximum gap configuration for the analysis .....	67
Figure 3.17.	Identification of the topology for each part of the block on rail assembly.....	69



Figure 3.18.	Representation of the assembly graph for the block on rail assembly.....	69
Figure 3.19.	Representation of the part graph for the rail .....	70
Figure 3.20.	Representation of the part graph for the block.....	70
Figure 3.21.	Representation of the part graph for a screw .....	70
Figure 3.22.	The transformations of path 2 .....	72
Figure 3.23.	Determining the relative positioning between Axis13 and R7. ....	74
Figure 3.24.	Calculating the relative positioning between R7 and Axis13 .....	75
Figure 3.25.	Determining the relative positioning between G7 and G4.....	76
Figure 3.26.	Minimum gap calculation of the relative positioning between G7 and G4 (used in the maximum gap calculation of the control function) .....	76
Figure 3.27.	Maximum gap calculation of the relative positioning between G7 and G4 (used in the minimum gap calculation of the control function).....	77
Figure 3.28.	Prototype application of a 3D graphical tolerance analysis module in VR.....	79
Figure 3.29.	Part of the user interface of the prototype.....	79
Figure 4.1.	UML diagram representing the main classes of the tolerance library .....	87
Figure 4.2.	Components of a geometrical tolerance.....	88
Figure 4.3.	Kinematics joints defined in the library. A solid arrow indicates a DOF that is free or blocked. A dashed arrow indicates a partially blocked DOF.....	91
Figure 4.4.	The interpretation of the tolerances in the tolerance analysis process.....	92
Figure 4.5.	Identification of the topological elements of the block bolted on rail assembly.....	94
Figure 4.6.	Detail drawing of the rail with identification of the tolerances .....	94
Figure 4.7.	Representation of the three paths that close the control function between R7 and G4 on the block bolted on rail assembly .....	94
Figure 4.8.	Nomenclature of the Part Graph .....	97
Figure 4.9.	Representation of the part graph for the rail .....	97
Figure 4.10.	Dependencies for Axis13.....	98

Figure 4.11.	Dependencies for R7 .....	98
Figure 4.12.	Dependencies common to Axis13 and R7 .....	98
Figure 4.13.	Text file output for the control function R7 on rail to G4 on Slider .....	100
Figure 4.14.	Volume obtained using a prototype 3D graphical analysis method .....	101
Figure 5.1.	Exemple de maillage contenant des triangles dégénérés .....	105
Figure 5.2.	Exemple de maillage réparé .....	105
Figure 5.3.	Représentation des triangles dégénérés de la figure 5.51 .....	105
Figure 5.4.	Maillage avec triangle dégénéré au centre .....	106
Figure 5.5.	Maillage dont le triangle dégénéré a été supprimé .....	106
Figure 5.6.	Graphe de test .....	109
Figure 5.7.	Tous les chemins possibles qui commencent au nœud 0 .....	109
Figure 5.8.	Définition du fichier de définition d'un graphe d'assemblage avec pièces tolérancées .....	110
Figure 5.9.	Définition du fichier de sortie .....	112

## LISTE DES SIGLES ET ABRÉVIATIONS

BGL	Boost Graph Library
CAA	Component Application Architecture
CAO	Conception assistée par ordinateur
CAVE	Cave Automatic Virtual Environment
DDL	Degrés de liberté
DOF	<i>Degrees of freedom</i>
FT&A	<i>Functional Tolerancing and Annotations</i> (atelier de Catia V5)
GD&T	<i>Geometric dimensioning and tolerancing</i>
LMC	<i>Least Material Condition</i>
MMC	<i>Maximum Material Condition</i>

## LISTE DES ANNEXES

ANNEXE I :Règles de conception des tolérances indépendantes pour sept types de contraintes géométriques variationnelles indépendantes (SVGC).....	122
ANNEXE II :Règles de conception des tolérances dépendantes pour 27 types de contraintes géométriques variationnelles dépendantes (CVGC) .....	123
ANNEXE III :Fichier d'entrée pour l'assemblage du glissoir vissé sur rail.....	126
ANNEXE IV :Rapport d'interprétation des tolérances pour l'assemblage du glissoir vissé sur rail.....	129
ANNEXE V : Signification des codes numériques utilisés dans le fichier d'assemblage.....	145

## AVANT-PROPOS

L'objectif initial de cette recherche était de résoudre des difficultés liées au problème d'assemblage entre une cheville et son trou. La difficulté à assembler deux pièces est due au fait qu'il y a des erreurs dans la fabrication et dans la manipulation des pièces. Un des objectifs était de tenir compte des sources d'erreur lors des opérations d'assemblages afin de déterminer les commandes de manipulation ayant le plus de chance de succès.

Les travaux initiaux ont portés sur le développement de méthodes permettant de générer une représentation graphique de l'espace à l'intérieur duquel les surfaces tolérancées peuvent exister en considérant les tolérances et les erreurs d'assemblage. Ces espaces sont les volumes d'existence et il devait être possible de les visualiser dans un environnement de réalité virtuelle de type CAVE. L'application à développer devait permettre à l'utilisateur de modifier les annotations géométriques et dimensionnelles et de constater le résultat sur les volumes d'existences. Les travaux ont beaucoup progressés dans cette direction. Par contre, plusieurs infrastructures essentielles nécessitaient d'être implémentées et d'autres problèmes résolus avant de compléter le module de génération des volumes d'existence. L'élément principal qui manquait était l'interprétation automatique du langage de tolérancement. C'est sur ce sujet que les efforts ont été mis par la suite.

Ce mémoire élabore sur l'état des travaux actuels sur l'analyse des tolérances, la plateforme de simulation des volumes d'existence dans l'environnement de réalité virtuelle qui fut développée dans le cadre de cette recherche, l'automatisation de l'interprétation des tolérances, et la direction que devraient prendre les travaux futurs.

## INTRODUCTION

Les logiciels de conception assistée par ordinateur (CAO) disponibles actuellement sur le marché sont très utiles pour concevoir et analyser des pièces et des produits. Malgré le perfectionnement des outils d'analyse par éléments finis, d'ergonomie, de cinématique, ainsi que tous ceux qui permettent de valider les différents aspects du produit avant la production, il existe toujours un vide important entre les phases de conception, d'industrialisation et de fabrication.

Un des problèmes est que les procédés manufacturiers introduisent des imperfections dans le produit. En effet, puisqu'il est impossible de fabriquer une pièce à une grandeur exacte, il est nécessaire d'atteindre un degré de précision qui assure la fonctionnalité du produit. C'est la raison pour laquelle des tolérances sont appliquées sur chaque cote. Les concepteurs doivent indiquer le degré de précision requis sur la pièce, en tenant compte des erreurs de fabrication, pour mener à la fonctionnalité du produit.

De plus, puisque les compagnies ont recours à la sous-traitance, il est nécessaire d'adopter le concept de la fabrication interchangeable. C'est-à-dire que les pièces qui sont produites dans des endroits très distants doivent pouvoir s'assembler lorsqu'on les réunit. Il faut donc s'assurer que toutes les pièces d'un même groupe soient assez semblables pour être interchangeables.

L'objectif de la procédure de tolérancement est de contrôler la qualité tout en minimisant les coûts d'opérations. Cependant, il n'y a pas de méthode universelle qui dicte la façon d'appliquer les tolérances et, dans la pratique, les concepteurs ont tendance à mettre des tolérances plus serrées que nécessaire, ou bien ils négligent de tolérer une surface fonctionnelle. Par conséquent, il en résulte des coûts plus élevés à cause de l'excès de précision demandé. Également, si on se rend compte que certaines tolérances sont inappropriées à un stade avancé dans le processus de développement ou de production, alors le taux de rejet et les coûts associés à des actions correctives sont élevés.

Finalement, il est difficile d'effectuer à la main l'analyse de tolérance sur une pièce en trois dimensions dès que sa géométrie est moindrement complexe. Il y a donc un besoin pour des logiciels d'aide au tolérancement.

L'objectif de cette recherche est de contribuer à la résolution du problème de l'analyse des tolérances en 3D. Le problème est divisé en deux tâches : l'interprétation des tolérances et la résolution des fonctions de contrôle. À cette fin, une méthode pour l'interprétation automatique des annotations existantes sur les pièces d'un assemblage et une nouvelle méthode graphique pour résoudre les fonctions de contrôle sont proposées.

L'interprétation des tolérances permet de construire un modèle topologique de chaque pièce tolérancée et un modèle de l'assemblage. Ces structures de données permettent de capter la signification des annotations et facilite la résolution de la fonction de contrôle.

L'analyse des tolérances par approche graphique est une extension des méthodes graphiques 2D existantes. Le résultat est un volume d'existence qui représente l'espace 3D où les surfaces à analyser pourraient exister. Un des avantages provient du fait que ce volume est un polygone qui peut être visualisé et analysé par des algorithmes de détection de collision et de pénétration afin de calculer les jeux ou serrages avec d'autres surfaces.

Le chapitre 1 présente un complément de revue de littérature pour cette recherche. Dans le cadre de ce mémoire par article, certains sujets plus précis de la revue de la littérature sont aussi présentés concernant la résolution des tolérances au chapitre 3 et concernant l'interprétation des tolérances au chapitre 4.

Le chapitre 2 présente l'application qui a été développée pour faire l'analyse des tolérances par l'approche graphique 3D. Cela constitue le premier volet de ce mémoire. Les objectifs, les outils informatiques utilisés, les interfaces utilisateurs et la méthodologie pour l'analyse des tolérances avec les volumes d'existence sont présentées.

Les chapitres 3 et 4 portent sur l'automatisation du processus de l'interprétation des tolérances, ce qui forme le deuxième volet de cette recherche. Le chapitre 3 présente la méthode d'interprétation des tolérances. On y présente un exemple d'assemblage commun dans l'industrie et un problème d'analyse des tolérances est présenté. Ce problème est résolu par une méthode traditionnelle 1D couramment utilisée dans l'industrie et ensuite par la méthode d'interprétation des tolérances qui est proposée dans cette recherche. Les deux méthodes sont ensuite comparées. Le chapitre 4 présente le déploiement de la librairie. L'interprétation des tolérances est possible grâce aux structures de données utilisées pour représenter les pièces et les assemblages et grâce aux algorithmes qui permettent d'obtenir les boucles cinématiques et les tolérances qui affectent chaque élément topologique. Cette solution est présentée sous forme d'une librairie C++.

La discussion générale au chapitre 5 présente les difficultés rencontrées, par rapport à la génération du volume des zones de tolérances, qui ont mené au développement de la librairie pour l'automatisation de l'interprétation des tolérances. Les résultats obtenus concernant l'interprétation des tolérances sont aussi discutés dans ce chapitre. Finalement, la conclusion fait part de la direction que devraient prendre les travaux futurs.



## CHAPITRE 1 : REVUE CRITIQUE DE LA LITTÉRATURE

Ce présent chapitre présente un aperçu des principaux logiciels commerciaux qui existent pour analyser les tolérances ainsi qu'un résumé des différentes techniques reliées à l'analyse des tolérances.

### 1.1 Logiciels commerciaux pour l'analyse des tolérances

#### 1.1.1 Catia V5

Le logiciel Catia V5 de Dassault Systèmes possède l'atelier *Functional Tolerancing and Annotations* (FT&A) qui permet d'appliquer les cotations et autres annotations directement sur le modèle solide. Le but de cet atelier est d'utiliser davantage une représentation des cotations directement sur le modèle maître plutôt que de générer des dessins 2D. La Fig. 1.1 illustre un exemple de pièce sur lequel sont appliquées des annotations 3D.



Figure 1.1. Aperçu de l'atelier *Functional Tolerancing and Annotations* (FT&A) de Catia V5 [1]

Les annotations sont associées à un ou plusieurs éléments géométriques du modèle 3D et il est possible de modifier cette liste manuellement. Par exemple, une tolérance de positionnement peut être associée à un trou ou à plusieurs trous. Cette fonctionnalité est particulièrement intéressante dans un contexte d'analyse des tolérances car la topologie associée à chaque annotation peut être récupérée par programmation avec Component Application Architecture (CAA).

De plus, l'atelier répond aux nouvelles normes de cotations 3D telle que définie par ASME Y14.41-2003. Il assure donc un certain contrôle automatique sur l'approche de cotation. Les annotations peuvent être modifiées sur un certain nombre de propriétés qui répondront toujours à la syntaxe de la norme choisie. Il est possible de générer un rapport sur le respect des règles de cotations, mais en bout de ligne, la cotation et le choix des dimensions sur le tolérancement dépend du concepteur. Cet assistant garantit seulement la validité de la syntaxe et de la sémantique du tolérancement. Cet atelier ne permet pas d'optimiser les tolérances. Pour plus de détails sur cette application, le lecteur peut consulter le site web de Dassault Systèmes [1].

### **1.1.2 TASYSSWorks**

TASYSSWorks est un logiciel de tolérancement intégré au logiciel de CAO SolidWorks de Dassault Systèmes. Selon ce système, il y a trois phases pour appliquer, analyser et optimiser le tolérancement. Ces phases correspondent respectivement aux applications : PreTASYSSWorks, TASYSSWorks et TASYSSWorks Intol.

Dans PreTASYSSWorks, l'utilisateur crée la fonction de contrôle, ce qui correspond à une mesure entre deux éléments pour lequel le jeu doit être vérifié. Il définit ensuite les limites supérieures et inférieures à respecter. Le logiciel recherche alors toutes les cotations qui sont susceptibles d'affecter la fonction de contrôle. L'utilisateur ajoute les cotes de tolérancement manquantes ou nécessaires. À ce niveau, il est possible de

visualiser le produit, et plus précisément, la fonction de contrôle, à sa valeur nominale, minimale, maximale et médiane.

Pour une analyse plus approfondie du tolérancement, il faut utiliser TASYSSWorks. Cet atelier permet d'étudier, de façon statistique ou déterministe, l'effet des annotations sur la fonction de contrôle. L'approche statistique assume une distribution normale autour de la valeur nominale pour les annotations qui ont un effet sur la fonction de contrôle. On calcule ainsi la distribution de la fonction de contrôle ce qui permet de donner un indice sur le pourcentage de pièces qui seront non-conformes. L'analyse déterministe calcule la combinaison de valeurs qui génèrera les résultats extrêmes sur la fonction de contrôle. Il est possible de voir la contribution et la sensibilité de chaque cotation sur l'erreur de la fonction de contrôle. La sensibilité est un indice de la variation qui résultera sur la fonction de contrôle si la valeur de la cote en question est changée. Prenant en considération ces informations, l'utilisateur change manuellement les tolérances en resserrant celles qui contribuent le plus à l'erreur et qui sont sensibles et il augmente les tolérances qui contribuent peu et qui sont moins sensibles. Il relance ensuite la série d'analyses et itère de cette façon jusqu'à ce que le résultat soit acceptable. Finalement, TASYSSWorks Intol est un module qui permet d'optimiser automatiquement les tolérances en prenant en considération d'autres critères comme les coûts de fabrication.

### **1.1.3 Cetol Six Sigma**

Cetol Six Sigma est un logiciel de tolérancement qui est un ajout à Catia V5 ou Pro/E. Le logiciel développé par Sigmetrix s'intègre directement dans Catia V5 grâce à un partenariat logiciel CAA V5 avec Dassault Systèmes. Les informations sur les annotations et le modèle 3D sont partagées avec Cetol Six Sigma qui crée son propre modèle de tolérancement. Le fonctionnement et les analyses obtenues sont similaires à celles obtenues par TASYSSWorks (graphes des contributions, graphe de sensibilité, paramètres statistiques  $C_{EK}$ , rapport d'analyse, etc). CETol Six Sigma utilisait une approche vectorielle pour résoudre l'analyse des tolérances. La version plus récente utilise une méthode variationnelle. Ces techniques sont décrites à la section 1.2.

#### **1.1.4 3DCS**

3DCS est un logiciel développé par Dimensional Control Systems, Inc. et il est disponible en deux versions : un logiciel indépendant et un logiciel qui est intégré dans Catia V5 grâce à un partenariat logiciel CAA V5 avec Dassault Systèmes. Similairement aux autres logiciels d'analyse des tolérances, 3DCS permet d'effectuer des analyses statistiques et obtenir un rapport de sensibilité et de contributions de l'erreur des annotations sur la fonction de contrôle. 3DCS utilise une approche variationnelle pour résoudre l'analyse des tolérances. Cette technique est décrite à la section 1.2.

#### **1.1.5 VisVSA**

VisVSA est un logiciel indépendant commercialisé par Unigraphics Solutions, Inc. (UGS). La géométrie doit être traduite en format «.jt» avant d'effectuer l'analyse des tolérances. Le format jt est une représentation tessellé du modèle 3D. VisVSA génère des rapports similaires à 3DCS et utilise aussi l'approche variationnelle dont le concept est décrit à la section 1.2.

### **1.2 État de la recherche**

L'intégration de l'analyse des tolérances à la phase de conception assistée par ordinateur (CAO) a fait l'objet de nombreuses recherches au cours des vingt dernières années. Plusieurs approches ont été développées et certaines ont été intégrées à des logiciels de CAO. Les méthodes d'analyse déterministes et statistiques sont très répandues dans les logiciels commerciaux mais il y a peu d'analyses graphiques. La différence se situe dans la façon dont les résultats sont obtenus et présentés. L'analyse déterministe vise à déterminer les cas extrêmes qui peuvent subvenir, par exemple le jeu minimum et le jeu maximum. Avec l'analyse statistique, une possibilité est que des distributions normales sont associées aux tolérances et le résultat indique la distribution de la fonction de contrôle et la probabilité que les spécifications ne soient pas respectées. Une autre possibilité consiste à générer une population de configurations possibles en faisant varier les paramètres du modèle de tolérance et étudier les résultats de façon statistique. Les

méthodes déterministes ou statistiques peuvent être accompagnées d'un rapport qui indique la sensibilité et la contribution de chaque annotation sur la fonction de contrôle. L'analyse graphique doit permettre de visualiser le volume d'existence, le volume dans lequel la zone de tolérance peut exister.

La façon générale d'effectuer l'analyse des tolérances consiste en les étapes suivantes:

- 1) Création du modèle numérique des pièces ;
- 2) Définition du processus d'assemblage ;
- 3) Application des tolérances ;
- 4) Identification des conditions fonctionnelles (les fonctions de contrôle) ; et
- 5) Analyse des résultats.

À la première étape, le modèle numérique des pièces sont créés aux dimensions nominales. Cela signifie que le modèle représente la condition parfaite de la pièce sans considération des erreurs de fabrication. La deuxième étape indique la façon dont les pièces forment des sous-assemblages. Des liaisons d'assemblages entre les pièces indiquent le positionnement relatif entre elles. La troisième étape est la définition des tolérances sur chaque pièce. L'analyse des tolérances prend du sens seulement si l'assemblage au complet est considéré car il permet de prendre connaissance tôt dans le processus de développement des jeux ou serrages qui peuvent se produire suite à un empilement des tolérances sur plusieurs pièces. La quatrième étape consiste à indiquer les endroits à contrôler en définissant des fonctions de contrôle. Par exemple, afin de connaître l'espacement possible entre la portière et le châssis, il faut définir une fonction de contrôle entre les deux pièces afin d'indiquer au logiciel que cette zone doit être analysée. Finalement, si le processus est automatisé, l'utilisateur lance l'analyse et le logiciel fournit des résultats que l'analyste doit vérifier.

Plusieurs approches ont été développées pour effectuer l'analyse des tolérances. Les principales catégories d'analyses des tolérances sont: 1D, 2D, et 3D. L'analyse 1D est

aussi appelée uniaxiale. Parmi les catégories d'analyse 2D et 3D, il y a les approches paramétrique, vectorielle et variationnelle. De plus, peu importe l'approche utilisée, l'analyse peut être de type déterministe ou statistique tel que discuté plus tôt.

De façon générale, l'analyse 3D est la plus désirable car elle tient compte des erreurs d'orientation en plus des erreurs de positionnement. À l'autre extrême, l'analyse 1D s'applique dans une seule direction à la fois et pour chaque tolérance, il y a deux cas extrêmes. Shen [2] a comparé ces différentes approches en indiquant que des résultats différents sont obtenus avec chacune. Néanmoins, chaque méthode possède ses avantages et ses inconvénients. Lin [3] a comparé des méthodes 1D, 2D et 3D puis a montré l'effet d'empilement des tolérances géométrique ainsi que l'effet d'amplification de l'erreur géométrique en fonction des dimensions de la pièce.

### **1.2.1 Analyse 1D**

L'analyse 1D permet de résoudre la fonction de contrôle en utilisant uniquement des cotes parallèles à un axe. La méthodologie de cette approche, tel que décrite par James. D. Meadows [4] implique la démarche suivante. D'abord, l'utilisateur définit une fonction de contrôle pour indiquer ce qu'il veut analyser (par exemple, le jeu entre deux pièces ou l'espace minimal entre deux éléments géométriques d'une même pièce). Deuxièmement, il faut identifier tous les facteurs (dimensions, tolérances, position) qui peuvent influencer la fonction de contrôle. Si il y a plusieurs pièces, l'analyste doit positionner les pièces dans la configuration extrême qu'il souhaite étudier de façon à minimiser ou maximiser la fonction de contrôle. Ensuite, une boucle est définie et la direction positive est déterminée par convention. Les cotes de tolérancement géométrique doivent être intégrées à leurs éléments respectifs de la boucle. Finalement, en exprimant chaque cote dimensionnelle de la boucle en terme d'une moyenne plus ou moins sa tolérance bilatérale, il est aisé de calculer la dimension moyenne de la fonction de contrôle ainsi que sa tolérance. La dimension moyenne de la fonction de contrôle est la somme des dimensions moyennes des autres éléments de la boucle (certains éléments

sont négatifs selon la convention établie). De même, la tolérance bilatérale de la fonction de contrôle est la somme des autres tolérances dans la boucle en valeur absolue. Habituellement, l'utilisateur utilise une charte des tolérances intégrée dans un chiffrier pour effectuer ces calculs. Dans ce cas, la valeur nominale de chaque cote, sa direction positive ou négative, et les tolérances qui se rattachent à chaque surface de la boucle sont entrées dans la charte de tolérance. Les calculs de sommations sont effectués manuellement ou par le chiffrier.

La popularité de cette méthode en industrie relève de sa simplicité d'utilisation. Toutefois, puisque c'est une méthode manuelle, l'analyste doit posséder une bonne compréhension de toutes les règles de tolérancement. Donc, le risque d'erreurs d'interprétation est accru. De plus, les résultats obtenus sont limités à une dimension.

### **1.2.2 Analyse 2D**

L'analyse 2D permet d'étudier la fonction de contrôle en utilisant les cotes situées dans des plans parallèles incluant les cotes d'orientations autour de l'axe normal à ces plans. Le processus d'analyse est similaire à celui de l'analyse 1D. Il faut d'abord identifier les chaînes cinématiques, puis les analyser. Bien qu'il soit possible d'effectuer les calculs manuellement en utilisant les règles de trigonométrie, cette tâche est beaucoup plus fastidieuse que pour l'analyse 1D.

L'analyse paramétrique et l'analyse vectorielle sont des moyens d'automatiser le processus. Certains modeleurs de CAO permettent de définir le modèle nominal par un ensemble de contraintes. Cette approche permet de modifier rapidement le modèle en changeant les valeurs des paramètres. L'analyse paramétrique consiste à récupérer le modèle nominal construit par contraintes et allouer à chaque contrainte un paramètre de tolérance. Cette approche était surtout utilisée dans les applications de CAO 2D. Le problème avec cette approche est que les tolérances appliquées se limitent aux contraintes de conception qui sont définies. Les contraintes du modèle ne correspondent pas

forcément aux annotations et aux tolérances qui doivent être contrôlées lors de la fabrication. De plus, il est difficile de transposer les tolérances selon les contraintes de constructions. Il est donc préférable de construire un modèle de tolérancement indépendant du modèle nominal. D'ailleurs, Kandikjan et Shah [5] ont précisé qu'il est préférable de ne pas confondre le modèle nominal paramétrique avec le modèle de tolérancement, car les deux ont un objectif différent. On peut donc argumenter que l'analyse 1D est avantageuse par rapport à l'analyse paramétrique 2D car elle permet de considérer tous les types de tolérances permises par les normes.

L'alternative pour poursuivre l'analyse des tolérances en 2D est d'utiliser une méthode de résolution vectorielle. Dans ce cas, il faut définir les chaînes cinématique 2D qui complètent la fonction de contrôle à partir d'une section de coupe, entrer les valeurs de chaque vecteur ainsi que sa tolérance dans le logiciel et la fonction de contrôle est calculée automatiquement par le logiciel.

### **1.2.3 Analyse 3D**

L'analyse 3D permet d'étudier la fonction de contrôle en utilisant toutes les cotes selon trois directions orthogonales, ainsi que selon les trois orientations, soit un total de 6 degrés de liberté. Le but de l'analyse 3D est d'obtenir un résultat qui se rapproche encore plus de la réalité qu'il ne l'est possible par les analyses 1D et 2D. De plus, le modèle de tolérancement peut être construit en réutilisant directement les modèles de pièces nominales, d'assemblage et les annotations qui sont définies lors des étapes précédentes du design. Cela évite la nécessité de créer un modèle indépendant de plus. On retrouve deux approches principales pour résoudre l'analyse des tolérances en 3D : soit l'approche variationnelle et l'approche cinématique (ou vectorielle 3D).

L'approche variationnelle consiste à varier le positionnement et l'orientation des éléments de la pièce lors de l'analyse. Pour ce faire, un modèle abstrait est construit à partir du modèle nominal, dans lequel les principales caractéristiques de la pièce originale



sont représentées par des primitives géométriques. Des variables sont associées à chaque degré de libertés de la zone de tolérance afin de les contraindre par rapport aux autres primitives du modèle abstrait. L'approche variationnelle 3D s'apparente à l'approche paramétrique 2D décrite précédemment sauf que dans l'approche paramétrique 2D, les tolérances étaient appliquées sur les contraintes déjà existantes du modèle nominal. Dans l'approche variationnelle 3D, il y a une distinction nette entre le modèle nominal et le modèle utilisé pour le tolérancement. Le modèle abstrait permet de créer de nouvelles contraintes qui représentent mieux la signification des tolérances.

Les logiciels commerciaux tel 3DCS, VisVSA et CETol Six Sigma utilisent l'approche variationnelle. La première étape est de représenter les caractéristiques de la pièce par des entités de base, tel un point, une ligne, un cercle, un cylindre, etc. Chaque entité de base est définie par des paramètres. Par exemple, un point est représenté par un vecteur position, une ligne est représentée par deux points, un cercle par un point, un rayon et un plan, etc. L'interprétation des tolérances consiste à définir des contraintes entre les entités de bases selon les tolérances qui sont définies. Les contraintes possibles sont définies à la Fig. 1.2. Une analyse statistique Monte Carlo est effectuée à partir du modèle abstrait. Schlatter [8] décrit comment une version initiale du logiciel VSA a été utilisée pour mener l'analyse de tolérances sur les lecteurs de disquettes de IBM. L'article décrit les bénéfices du logiciel et donne un aperçu de la démarche et les résultats obtenus, mais sans entrer dans les détails des algorithmes à la base de l'application. Shen [9] a évalué une version plus récente du logiciel VisVSA. Selon sa critique, VisVSA est un outil utile pour comprendre les causes de variations géométriques dans les pièces et les assemblages. Toutefois, certaines lacunes sont identifiées. D'abord, dû au fait que les modèles sont importés dans VisVSA, il y a une perte d'information dans le transfert des données. Toutes les contraintes géométriques et dimensionnelles sur les pièces, l'historique de construction ainsi que les contraintes d'assemblage sont perdues et doivent être recréées dans le logiciel. Cela équivaut à une redondance dans le travail mais n'enlève rien à la qualité de l'analyse. D'ailleurs, au moment d'écrire ce texte, des

travaux sont en cours par les développeurs de VisVSA pour permettre l'utilisation d'un format de fichier qui permettrait de récupérer ces données. La deuxième critique porte sur la validation des spécifications de contrainte qui pourrait être améliorée. Par exemple, certaines règles sont appliquées afin que l'utilisateur applique les annotations en conformité avec les normes établies (ASME, ISO), mais aucune validation n'est effectuée sur les référentiels afin d'assurer qu'ils contraignent tous les degrés de libertés. En troisième lieu, on soupçonne constamment les algorithmes derrière le logiciel puisque ceux-ci ne sont pas publiés. Le quatrième point relevé est que l'utilisation du logiciel requiert de l'utilisateur une expertise des normes de tolérancement en plus d'une expertise du logiciel. Par exemple, dans certain cas, il n'est pas clair quel type de fonction de contrôle doit être définie. Dans VisVSA, l'analyste doit spécifier le type de fonction de contrôle qu'il souhaite définir (point-point, point-ligne, point-plan, etc). Finalement, le logiciel fournit seulement des analyses statistiques. Il ne fournit pas d'analyse déterministe.

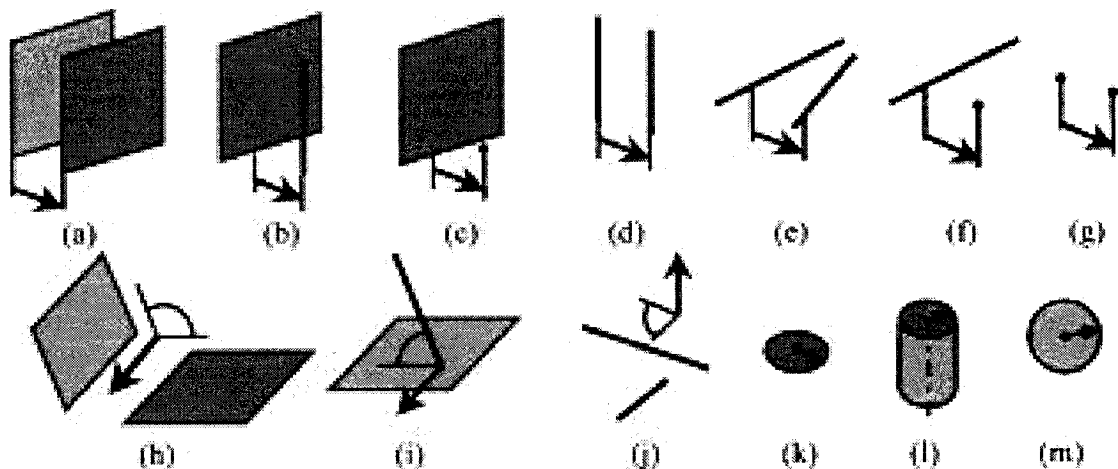


Figure 1.2. Relation entre les éléments du modèle paramétrique abstrait [2] : a) contrainte plan-plan parallèle ; b) contrainte plan-ligne parallèle ; c) contrainte plan-point ; d) contrainte ligne-ligne parallèle ; e) contrainte ligne-ligne co-planaire ; f) contrainte ligne-point ; g) contrainte point-point ; h) contrainte plan-plan non-parallèle ; i) contrainte plan-ligne non-parallèle ; j) contrainte ligne-ligne non-planaire ; k) contrainte de forme ; l) contrainte de cylindricité ; m) contrainte de circularité.

Une technique variationnelle alternative consiste à mailler la pièce et associer à chaque nœud les tolérances qui lui reviennent. Cette approche peut permettre de visualiser différents cas qui pourraient se présenter. Gupta [6] présente un moyen de visualiser individuellement différentes possibilités d'une surface. Par exemple, pour une surface planaire à laquelle une tolérance est appliquée, il faut utiliser une représentation de surface de type *nurb*. Par contre, il n'y a pas de calculateur de contraintes et il ne propose pas de moyen d'interpréter les tolérances. Sa recherche se situe dans la représentation de pièces telles que produites. Pierce [7] s'est intéressé à générer plusieurs cas possibles de pièces produites et à simuler les contacts entre les surfaces fonctionnelles imparfaites. Sa recherche porte spécifiquement sur les tolérances de forme et d'orientation. L'objectif est d'allouer les tolérances qui permettent aux surfaces des liaisons d'assemblage de respecter certains critères géométriques associés au fonctionnement de la liaison d'assemblage.

L'approche cinématique est la principale alternative à l'approche variationnelle. Cela consiste à créer des chaînes cinématiques 3D qui permettent de résoudre la fonction de contrôle. Cette méthode est inspirée des théories de la robotique. Une chaîne cinématique doit passer d'une pièce à l'autre par l'intermédiaire des liaisons d'assemblages. Les chaînes cinématiques sont généralement obtenues par utilisation d'un graphe d'assemblage. Les erreurs sont considérées dans les liaisons d'assemblages et dans les annotations appliquées sur les pièces. Il existe plusieurs méthodes pour résoudre la fonction de contrôle.

Le processus de l'analyse cinématique se divise en deux grandes étapes : l'interprétation des tolérances et la résolution de la fonction de contrôle. L'interprétation consiste à identifier les chaînes cinématiques d'une fonction de contrôle et les tolérances qui influencent le positionnement et l'orientation pour chaque élément de la chaîne. La

méthode de résolution vise à résoudre la chaîne cinématique et à donner un résultat afin d'indiquer le jeu.

Puisque les méthodes de résolution par une approche cinématique ne sont pas le premier intérêt de cette recherche, ce sujet sera abordé en premier. Desrochers, Ghie et Laperrière ont proposé le modèle du jacobien-torseur unifié pour l'analyse des tolérances [10-14]. Les zones de tolérance des éléments fonctionnels sont exprimées sous formes de torseurs variationnels. Une matrice jacobienne (J) est définie afin de représenter les tolérances qui sont exprimées dans le repère des éléments fonctionnels (FEs), dans le repère de la fonction de contrôle (FR). La formule s'écrit ainsi :

$$[FR]=[J][FE_S] \quad (1.1)$$

[FR] : torseur pour représenter l'intervalle de variation de la fonction de contrôle, exprimé dans le repère de base;

[FE<sub>S</sub>] : regroupe les torseurs qui représentent la tolérance sur les éléments tolérancés de la boucle cinématique, les torseurs sont exprimé dans le repère local de chaque élément;

[J] : La matrice jacobienne permet d'exprimer les tolérances [FE<sub>S</sub>] dans le repère de base ;

Ou en plus de détails :

$$\begin{bmatrix} [u, \bar{u}] \\ [v, \bar{v}] \\ [w, \bar{w}] \\ [\alpha, \bar{\alpha}] \\ [\beta, \bar{\beta}] \\ [\delta, \bar{\delta}] \end{bmatrix}_{FR} = \begin{bmatrix} [J_1 & J_2 & J_3 & J_4 & J_5 & J_6]_{FE1} \dots [J_1 & J_2 & J_3 & J_4 & J_5 & J_6]_{FEN} \end{bmatrix} \begin{bmatrix} [u, \bar{u}] \\ [v, \bar{v}] \\ [w, \bar{w}] \\ [\alpha, \bar{\alpha}] \\ [\beta, \bar{\beta}] \\ [\delta, \bar{\delta}] \end{bmatrix}_{FE1} \dots \begin{bmatrix} [u, \bar{u}] \\ [v, \bar{v}] \\ [w, \bar{w}] \\ [\alpha, \bar{\alpha}] \\ [\beta, \bar{\beta}] \\ [\delta, \bar{\delta}] \end{bmatrix}_{FEN} \quad (1.2)$$

Où :

$[u, \bar{u}], [v, \bar{v}], [w, \bar{w}]$  : représente les limites inférieures et supérieures pour les 3 DDL de translation ;

$[\alpha, \bar{\alpha}], [\beta, \bar{\beta}], [\delta, \bar{\delta}]$  : représente les limites inférieures et supérieures pour les 3 DDL de rotation ;

$$\begin{bmatrix} [u, \bar{u}] \\ [v, \bar{v}] \\ [w, \bar{w}] \\ [\alpha, \bar{\alpha}] \\ [\beta, \bar{\beta}] \\ [\delta, \bar{\delta}] \end{bmatrix}_{FR}$$
 : représente l'intervalle de variation de la fonction de contrôle ;

$$\begin{bmatrix} [u, \bar{u}] \\ [v, \bar{v}] \\ [w, \bar{w}] \\ [\alpha, \bar{\alpha}] \\ [\beta, \bar{\beta}] \\ [\delta, \bar{\delta}] \end{bmatrix}_{FEi}$$
 : représente l'intervalle de variation pour le ième élément tolérancé (une zone de tolérance) ;

$[J_1 \ J_2 \ J_3 \ J_4 \ J_5 \ J_6]_{FEi}$  : la ième matrice jacobienne 6x6 qui permet d'exprimer le ième élément tolérancé dans le repère de base (voir [12-14] pour développement).

Les torseurs FEi expriment l'intervalle de variation des zones de tolérances sur chaque degré de libertés. La multiplication de la matrice jacobienne (J) par les torseurs d'éléments fonctionnels (FEi) exprime la contribution de la tolérance dans le repère de la fonction de contrôle (FR). La somme de toutes ces contributions indique la variation possible de la fonction de contrôle (FR). Finalement, il est spécifié que cette formule pourrait être utilisée pour la synthèse de tolérance en pré-multipliant chaque coté de l'équation par la matrice pseudo-inverse de la jacobienne. Le problème se formulerait alors de la façon suivante : quelles valeurs devraient avoir les autres tolérances afin d'assurer que la fonction de contrôle soit respectée.

Une autre approche de résolution est de formuler l'expression matricielle à partir des chaînes cinématiques et utiliser une expression en série de Taylor d'ordre un pour linéariser l'équation. Une telle approche a été utilisée dans les premières versions du logiciel CETol Six Sigma mais aurait par la suite été remplacée par une approche variationnelle selon Shen [2]. Cette approche a été développée par Chase à travers le programme ADCATS. Pour plus de détails sur cette approche de résolution, le site de l'ADCATS contient beaucoup d'informations pertinentes [15].

Un aperçu des méthodes de résolution variationnelle et cinématique a été présentée. Le constat est qu'avant de procéder à la résolution d'une fonction de contrôle, il est nécessaire d'interpréter les tolérances. L'automatisation de l'interprétation des tolérances est rendue possible en partie grâce à plusieurs structures de données en graphe, une pour l'assemblage et une pour chaque pièce. L'utilisation du graphe d'assemblage pour générer les chaînes cinématiques est un concept utilisé par plusieurs. Ce graphe décrit les liaisons d'assemblage qui existent entre les pièces. Dans tous les cas, le graphe d'assemblage a une forme et un but similaire. Ce n'est pas le cas pour le graphe de pièce. Dans la recherche de Desrochers [11], le graphe d'assemblage est utilisé pour définir les surfaces associées technologiquement et topologiquement (SATT) pour la synthèse des tolérances, alors que le graphe de pièces n'est pas utilisé.

Dans la théorie des SATT, chaque nœud du graphe représente une pièce. Les arêtes lient les surfaces fonctionnelles adjacentes de chaque pièce. En utilisant la théorie des graphes, une base de cycles indépendants est définie et utilisée afin d'associer les surfaces fonctionnelles d'une même pièce entre elles et ainsi former un repère sur la pièce.

#### **1.2.4 Le graphe de pièce**

Requicha a développé le concept du graphe variationnel (VGraph) en 1986 afin de représenter les tolérances [16]. Moroni et Requicha [17] ont implanté le VGraph afin

d'enregistrer les tolérances. La structure développée permet de supporter un modèle de tolérancement sur un modèle nominal en cours de développement. Au sommet du graphe se trouve les *features* (groupe d'éléments géométriques qui forment une partie distincte de la pièce) qui pointent vers les faces qui les composent. Si un *feature* ou les faces sont tolérancées, elles pointent vers une classe d'attributs qui contient le type de tolérance et la valeur de celle-ci. Les tolérances qui ont des références pointent vers un objet de type référentiel qui pointe à son tour vers d'autres faces. Bien qu'une relation existe entre les surfaces et les tolérances, la recherche ne laisse pas entrevoir une méthode d'analyse des tolérances.

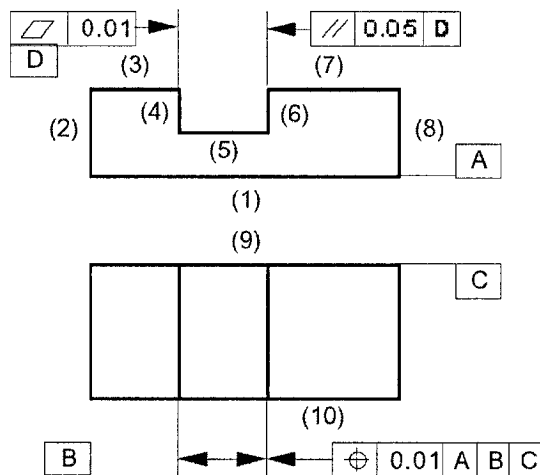


Figure 1.3. Dessin de détail partiel d'une pièce [17]

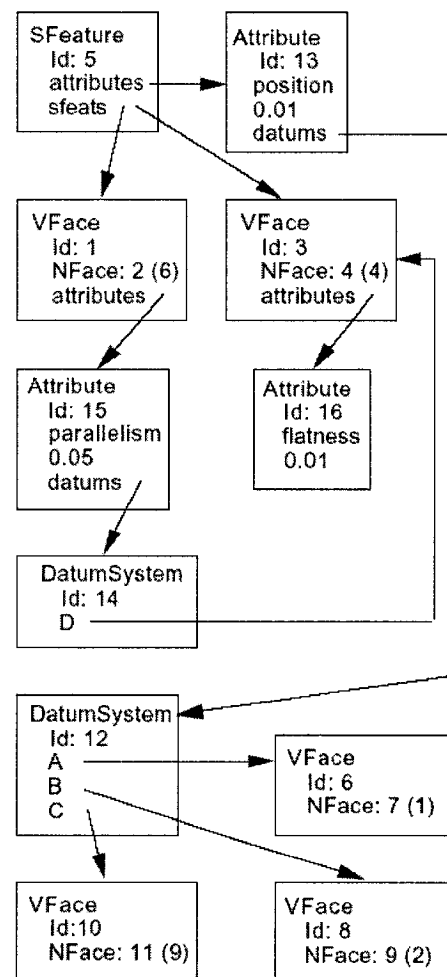


Figure 1.4. Fragment de VGraph [18]

Un exemple de pièce et de VGraph est présenté dans les Fig. 1.3 et 1.4. Le nœud *SFeature* représente la rainure dans la pièce. Le nœud *SFeature* pointe vers les deux faces intérieures de la rainure, soit les faces identifiées sur le dessin de détail par les chiffres 4 et 6. Dans le VGraph, ceci correspond au nombre en parenthèse dans les deux nœuds *VFace* pointés par le nœud *SFeature*. Une tolérance de planéité est appliquée sur la face 4, ce qui est représenté dans le Vgraph puisque le nœud de la *VFace* correspondante pointe vers un nœud *Attribute* de type planéité. La face 6 a une tolérance de parallélisme par rapport à la référence D (qui est aussi la face 4). Dans le VGraph, le nœud *VFace* correspondant à la face 6 pointe vers un nœud *Attribute* de type parallélisme, qui lui pointe vers un nœud *DatumSystem* qui pointe sur le nœud de la face 4. Finalement, le même raisonnement peut être appliqué pour la tolérance de positionnement qui est appliqué sur la rainure.

Ngoi [18] a développé un graphe de pièce afin de créer un outil qui facilite les calculs de transfert de cotes dans les pièces. Plutôt que de laisser à l'utilisateur le soin de remplir la charte de tolérance pour réaliser une analyse 1D, Ngoi construit le graphe de pièce et un algorithme détecte automatiquement la chaîne qui boucle la fonction de contrôle. La Fig. 1.5 illustre un exemple de pièce tolérancée. La Fig. 1.6 montre l'identification de chaque surface et axe de la pièce. La Fig. 1.7 représente le graphe de pièce. Les nœuds rectangulaires représentent les surfaces ou axes de la pièce. Les nœuds circulaires représentent les surfaces ou axes qui sont associés à des tolérances géométriques. Les arêtes représentent les distances entre les nœuds. Une arête pointillée représente une tolérance géométrique. Le sens des arêtes est décidé selon la convention choisie. Dans cet exemple, le sens positif est vers le bas. Ce graphe est utile pour sélectionner, ou aider à sélectionner, les annotations à considérer dans un calcul de transfert de cote. La technique peut aussi être utilisée pour faire une analyse 1D dans un assemblage. L'utilisateur doit positionner les pièces dans la configuration qu'il souhaite analyser, ajouter des lignes pointillées avec une distance nulle entre les surfaces en contact.



Cette méthode revient au même que d'utiliser une charte de tolérance pour faire l'analyse 1D. L'information sur le type de tolérance est perdue dans le graphe et il n'est pas clair comment une tolérance avec plus d'une référence serait représentée dans le graphe. D'autres auteurs ont proposé de construire un graphe pour chaque direction à analyser. Shah [19] proposait de créer un graphe pour les trois directions orthogonales puisque normalement trois chartes sont nécessaires pour les pièces prismatiques.

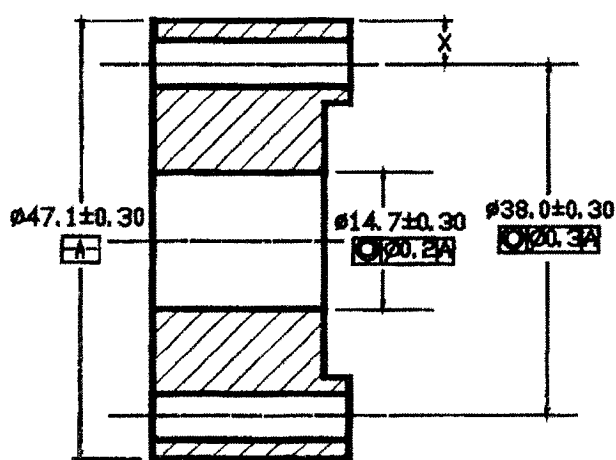


Figure 1.5. Pièce tolérancée [18]

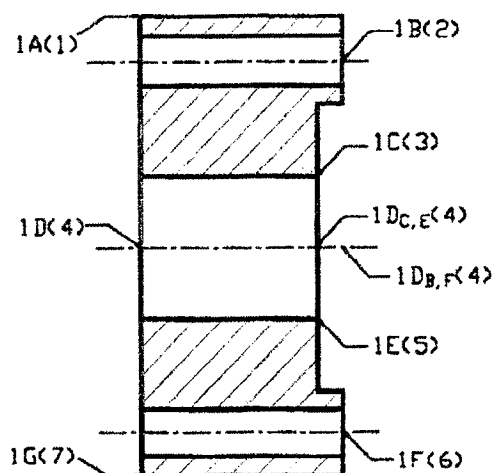


Figure 1.6. Identification de la topologie [18]

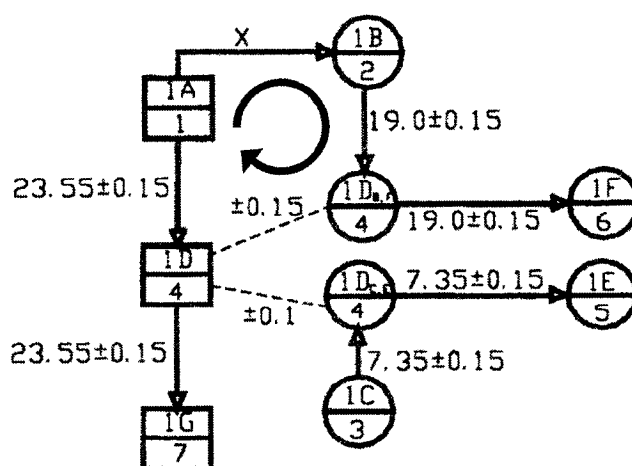


Figure 1.7. Graphe de pièce (légèrement modifié) [18]

Kandikjan et Shah [20] ont proposé une structure de graphe similaire afin de valider la syntaxe des tolérances appliquées sur un modèle de pièce 3D. Le graphe à la Fig. 1.9 représente le dimensionnement et le tolérancement de la pièce à la Fig. 1.8. Les nœuds sont des éléments géométriques tandis que les arêtes représentent les tolérances. Les arêtes sont directionnelles lorsqu'elles représentent des tolérances géométriques avec référence et elles sont bi-directionnelles pour les autres cas. Le graphe représente les tolérances explicites (celles qui sont cotées sur les dessins de détails) et implicites (celles qui ne sont pas cotées, par exemple si deux lignes sont perpendiculaires ou parallèles sur le dessin de détails, alors on assume une tolérance de perpendicularité ou de parallélisme correspondant à la tolérance indiquée dans la note générale). Quatre types de sous-graphes sont formés dans le graphe :

- $D \subset G$  : système de références utilisé pour spécifier des tolérances géométriques (*datum reference frame*, DRF);
- $P \subset G$  : sous-graphe utilisé pour des tolérances de positionnement appliquées à des patrons ;
- $Pr \subset G$  : sous-graphe pour les tolérances de profil ;
- $R \subseteq G$  : sous-graphe des *features* associées aux systèmes de références (T-DRF).

Le sous-graphe DRF regroupe les nœuds références, les relations géométriques et les dimensions de base entre elles. Le but de ce sous-graphe est de vérifier la validité du système de référence. Le sous-graphe de patron regroupe des ensembles d'éléments positionnés. Son but est de vérifier que ces éléments sont correctement contraints entre eux. Le sous-graphe T-DRF peut regrouper par exemple un sous-graphe DRF et un sous-graphe de patron. Il sert à valider que les éléments tolérancés sont correctement contraints par rapport à leur système de référence.

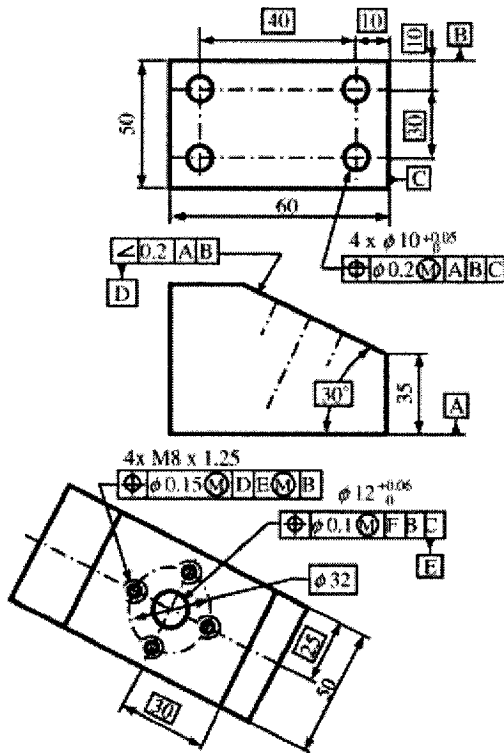


Figure 1.8. Pièce avec 4 systèmes de référence inter-reliés [20]

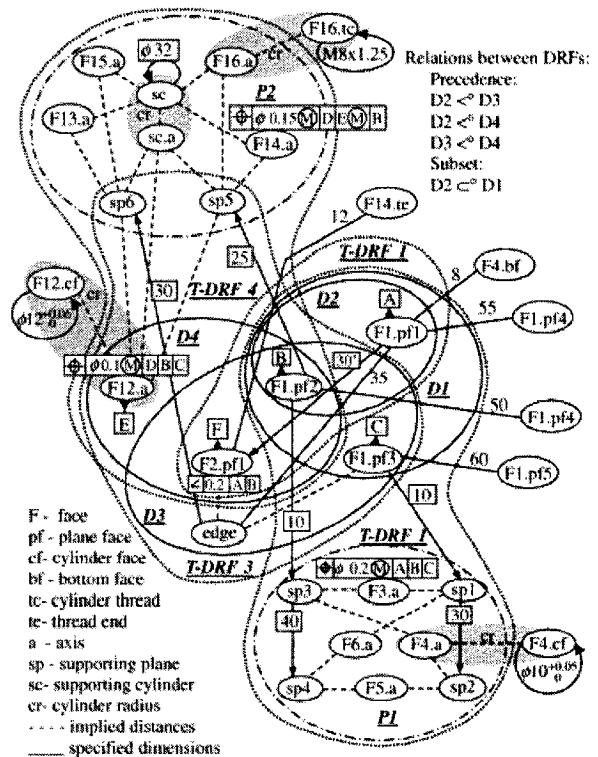


Figure 1.9. Graphe de la pièce à la figure 1.8 [20]

Kandikjan [20] propose aussi des contrôles sur le type de géométrie qui est associée à la tolérance géométrique. Par exemple, la validation qu'une tolérance de planéité est bien appliquée à une surface plane, ou qu'une condition de matière MMC est appliquée en référence à un *feature* de grandeur.

Les travaux de Hu [21] ont porté sur la synthèse des tolérances à partir d'un réseau de contraintes. Trois types de contraintes sont définis : les contraintes indépendantes, les contraintes dépendantes et les contraintes de contacts. Les deux premières font allusions aux types de tolérances qui peuvent être indépendantes si elles n'ont pas de référence, ou dépendantes dans le cas contraire. Les contraintes de contact servent à définir les degrés de liberté entre deux surfaces de pièce différentes qui sont en contact. Des règles sont utilisées pour définir les contraintes de type indépendantes ou dépendantes. Ces règles sont résumées dans les annexes I et II.

Hu a défini des règles pour construire le graphe des tolérances. La Fig. 1.10 montre un exemple d'assemblage non-tolérancé. La surface A5 sur la pièce mâle doit s'assembler avec la surface correspondante B5 sur la pièce femelle. Il en va de même pour les surfaces A4 et B4. Une déviation maximale de 0.017 mm est acceptable entre les surfaces A5 et B5. L'objectif est de tolérer la pièce en prenant en considération ces informations et les règles citées plus tôt.

Les couples A5-B5 et A4-B4 définissent des contraintes de contacts (*Mating Trees*, MT) sur le graphe à la Fig. 1.11. Ces contraintes de contact doivent être reliées entre elles en passant par les systèmes de référence de chaque pièce (voir DRF1 et DRF2 sur le graphe). Ainsi, une boucle est formée. La dernière étape consiste à définir des tolérances entre les éléments associés à A4, A5, B4 et B5 par rapport aux éléments composants le système de référence sur les pièces A et B. Les types de tolérances sont choisis en se référant aux règles des annexes I et II. Il reste donc à déterminer comment allouer une valeur à chaque tolérance. Ce problème est abordé dans [22] en utilisant un algorithme génétique.

Les graphes de pièces présentés ont tous été élaborés pour répondre à un besoin différent. Cependant, à la base ils ont tous une structure de graphe similaire qui est que les nœuds représentent les éléments topologiques de la pièce et les arêtes représentent les tolérances.

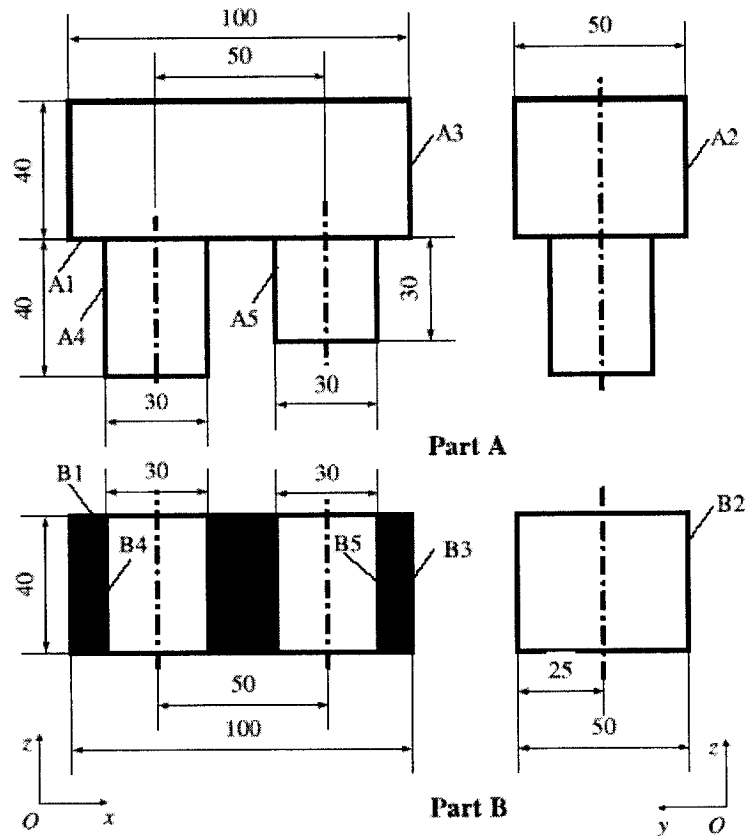


Figure 1.10. Exemple d'un assemblage non tolérancé [21]

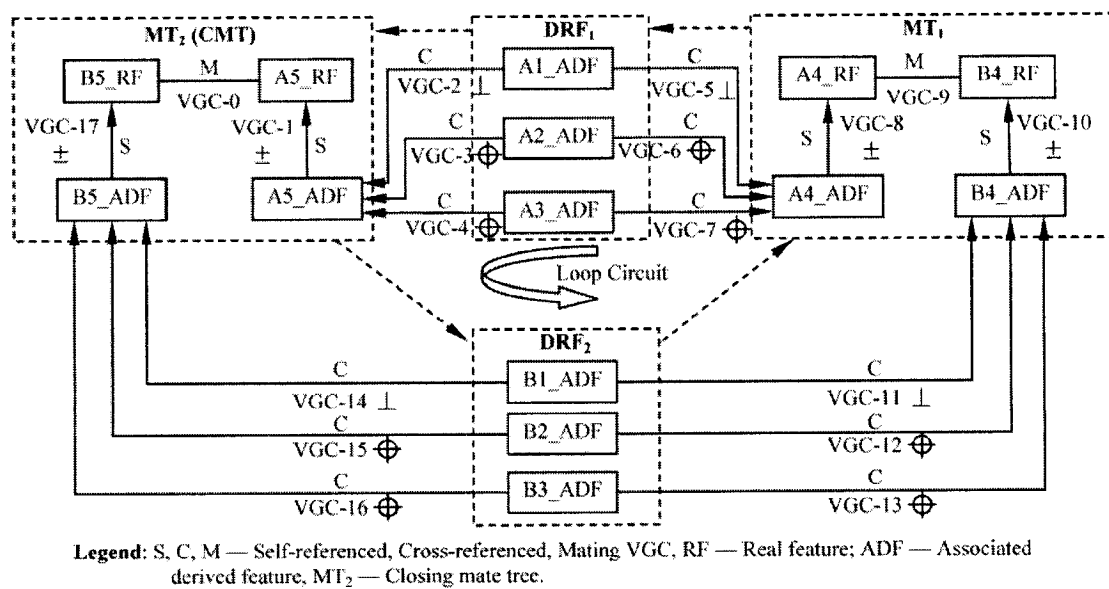


Figure 1.11. Le réseau de contraintes et le graphe des tolérances générées [21]

## **CHAPITRE 2: APPLICATION EN RÉALITE VIRTUELLE**

### **2.1 Démarche de l'ensemble du travail de recherche**

Cette recherche s'insère dans le cadre de l'utilisation de la réalité virtuelle pour la conception de produits. Un assistant à la création de cotes de tolérancement dimensionnel et géométrique a été développé et fonctionne dans un environnement de réalité virtuelle. L'utilisation de la réalité virtuelle est justifiée car elle permettra de tirer avantage de l'immersion de l'utilisateur et des sensations qui s'ajoutent pour fournir une meilleure compréhension du produit. À un stade plus avancé, il sera possible d'évaluer l'interaction humaine en réalité virtuelle dans un contexte de CAO. Il est certain qu'aujourd'hui le coût d'une telle utilisation est important, mais dans un proche avenir ces coûts décroîtront rapidement.

Le sujet de cette recherche, la génération d'une représentation graphique de l'espace à l'intérieur duquel les surfaces tolérancées peuvent exister, appelé volume d'existence, se prête bien à l'utilisation de la réalité virtuelle. L'avantage de la réalité virtuelle par rapport à une simple simulation sur un écran tient au fait que le concepteur peut ressentir un tolérancement au bout des doigts et ainsi en juger mieux l'impact sur le fonctionnement du produit. Cela permet aussi de se promener le long de la chaîne de cotes et ainsi visualiser le type de problème créé par un jeu ou un serrage non désiré. En travaillant à l'échelle des tolérances, il est parfois difficile de bien saisir l'effet des écarts lorsqu'un choix de tolérance entraîne des effets dans l'espace. L'outil peut aussi servir dans l'enseignement où il permet à l'étudiant de comprendre les notions parfois rébarbatives de jeu de chargement ou de références simulées lors de la conception d'un gabarit de contrôle, les notions de compliance en assemblage de produit ou les notions de répétabilité dans le respect de l'isostatisme lors de l'usinage, puis dans le contrôle d'une pièce.

Puisque cela consistait en un nouveau projet au sein du département, la démarche consistait à établir le plus rapidement possible la preuve de concept pour la génération

des volumes d'existence dans un environnement de réalité virtuelle. Il fallait développer la méthode pour générer les volumes d'existence, et ensuite l'implémenter dans l'environnement de réalité virtuelle. La génération des volumes d'existence fonctionne malgré certaines limitations.

L'avantage de cette approche scientifique est qu'elle a permis de valider rapidement le concept. L'application permet littéralement de voir les zones de l'assemblage où il y a des possibilités de collision entre les pièces. La tâche de l'utilisateur consiste à étudier les zones critiques et déterminer quelles tolérances il faut optimiser afin de minimiser la grandeur des volumes d'existence. Le désavantage qui résulte de cette approche scientifique est que le résultat final se limite à l'étude de quelques cas dû à des limitations dans l'application développées. En effet, il n'était pas possible d'intégrer toutes les subtilités du tolérancement afin de démontrer le concept des volumes d'existences. Une fois le concept démontré, il était nécessaire de revenir à la base pour automatiser l'étape de l'interprétation des tolérances afin de pouvoir mener l'étude plus loin.

La structure de ce document suit la chronologie de la démarche scientifique. C'est-à-dire que malgré le fait que l'analyse des tolérances soient divisée en deux étapes qui se suivent, (1) l'interprétation et (2) la résolution, dans ce mémoire c'est la résolution par les volumes d'existences qui est présentée en premier, suivi de l'interprétation des tolérances.

La résolution par les volumes d'existence est le sujet de ce chapitre. L'ensemble de cette recherche a aussi été une expérimentation avec la réalité virtuelle qui est également abordée dans ce chapitre. La réalité virtuelle est un moyen d'offrir à l'utilisateur une perspective différente de l'information. Une part importante des développements en réalité virtuelle est allouée à la création d'interfaces utilisateurs alternatives à ce que le moniteur traditionnel peut offrir. Il existe plusieurs systèmes qui permettent d'expérimenter avec la réalité virtuelle et chacun possède ses propres caractéristiques.

De façon générale, un des objectifs est de permettre à l'utilisateur d'utiliser des commandes plus intuitives ou naturelles grâce à une liberté de mouvement supérieure. Les outils utilisés, les outils développés, l'interface utilisateur et la théorie qui ont permis de démontrer le concept des volumes d'existences sont présentés.

Les chapitres 3 et 4 concernent l'interprétation des tolérances et l'application développée pour cette section est indépendante de l'application pour générer les volumes d'existences en réalité virtuelle. Le chapitre 5 présente une discussion générale sur les résultats et les limitations du travail dans son ensemble

## **2.2 Présentation des outils informatiques**

### **2.2.1 L'environnement de réalité virtuelle**

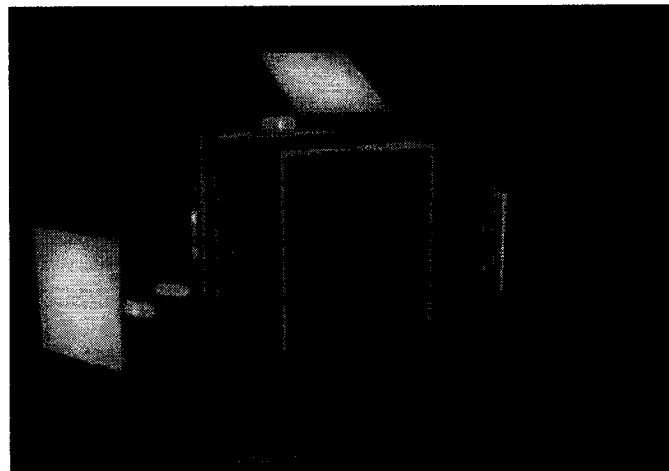


Figure 2.1. Description de la CAVE

Les résultats du travail présent permettent de tolérer un produit en temps réel. L'École Polytechnique de Montréal dispose d'un système de type CAVE (Cave Automatic Virtual Environment) [23]. La CAVE est une pièce cubique de 3 mètres de côté. Chaque mur ainsi que le plancher est un écran géant (tel qu'illustré à la Fig. 2.1) sur lequel est projeté l'image en stéréo. L'effet stéréo est créé par un affichage alternant pour les vues de l'œil gauche et de l'œil droit. Puisque les utilisateurs portent des



lunettes obturatrices qui se ferment en synchronisation avec l’affichage, l’œil gauche et l’œil droit perçoivent des vues différentes ce qui crée l’effet stéréo.

La CAVE est aussi dotée d’un système de repérage électromagnétique permettant de mesurer la position et l’orientation des capteurs. Ces capteurs sont fixés sur les lunettes de l’utilisateur et sur la souris 3D.

Une CAVE fonctionne normalement sur un système Onyx 2 de SGI, mais celle de l’école a été mise à jour et opère maintenant sur un réseau d’ordinateurs de bureau Pentium IV roulant chacun sous Linux Debian. Le réseau comporte un ordinateur maître, quatre ordinateurs pour l’affichage, un pour les données des capteurs, un qui sert de serveur et un poste de calcul.

L’ordinateur maître est utilisé pour lancer les applications. La communication avec les autres PC’s, l’interprétation des données et l’affichage en stéréo du code opengl est géré par l’intermédiaire des librairies vrjuggler et netjuggler [24].

### **2.2.2 Description des librairies utilisées pour l’application en réalité virtuelle**

Une décision critique au développement du projet fut sans aucun doute le choix de la librairie pour servir de fondement au projet. Parmi les éléments cruciaux à la réalisation du projet figurent l’utilisation d’une structure en graphe de scène et un « loader » de fichiers complets pour l’affichage d’objets complexes et texturés. Parmi les choix de librairies disponibles, Open Scene Graph [25] est celle qui semblait être la mieux développée, documentée et pour laquelle il est possible d’obtenir du support technique. Elle est accompagnée par de nombreuses classes pour effectuer du calcul matriciel, charger des modèles 3D ainsi que la plupart des formats d’images couramment utilisées, et elle facilite la création de la structure en graphe de scène.

### 2.2.3 Les menus

Il est important de noter que les outils informatiques existants pour travailler sur un environnement de réalité virtuelle sont extrêmement limités. Une classe de menus a été développée afin de permettre un plus grand nombre de fonctionnalités. Pour le moment, les menus disposent seulement de boutons poussoir et de fenêtres d’affichage de texte. De plus, ils sont définis dans le repère des lunettes 3D et donc apparaissent fixes par rapport à l’utilisateur. Par contre, ils peuvent être repositionnés n’importe où dans ce repère.

### 2.2.4 Structure du logiciel

La Fig. 2.2 illustre la relation entre les classes principales et les bibliothèques informatiques du logiciel.

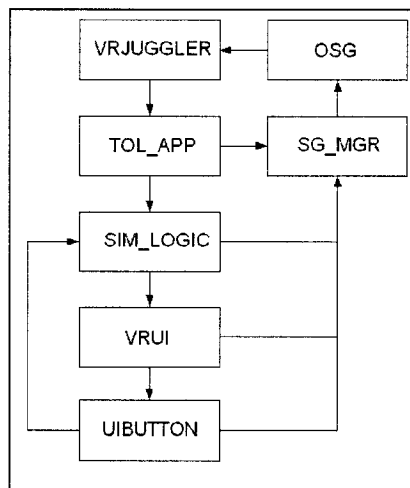


Figure 2.2. Structure du logiciel

La bibliothèque VrJuggler permet la communication avec l’interface des capteurs et gère l’affichage dans l’environnement de la CAVE. Les données des capteurs sont gérées dans le logiciel par la classe Tol\_App et elles sont utilisées par Sim\_Logic et SG\_MGR (« scene graph manager »).

Sim\_Logic gère les commandes de l'utilisateur en ayant recours aux données des capteurs et par l'utilisation des classes de menus, VRUI (Virtual Reality User Interface) et UIButton (User Interface Button). Au moment des tests en réalité virtuelle, la librairie pour l'interprétation du tolérancement n'avait pas encore été développée. Si elle l'avait été, elle aurait été utilisée par le module Sim\_Logic. Dans cette application, SIM\_Logic utilise une version préliminaire de la librairie d'interprétation des tolérances. Toutefois, la majorité des concepts utilisés étaient similaires (graphes de pièce et d'assemblage).

L'objet SG\_MGR gère le graphe de scène. Tous les éléments affichés dans la CAVE (menus, pièces) et les données des capteurs sont organisés dans le graphe de scène. La création du graphe de scène est possible grâce à la librairie Open Scene Graph. Finalement, c'est elle qui crée la liste d'affichage, mais c'est VrJuggler qui permet de l'afficher dans la CAVE.

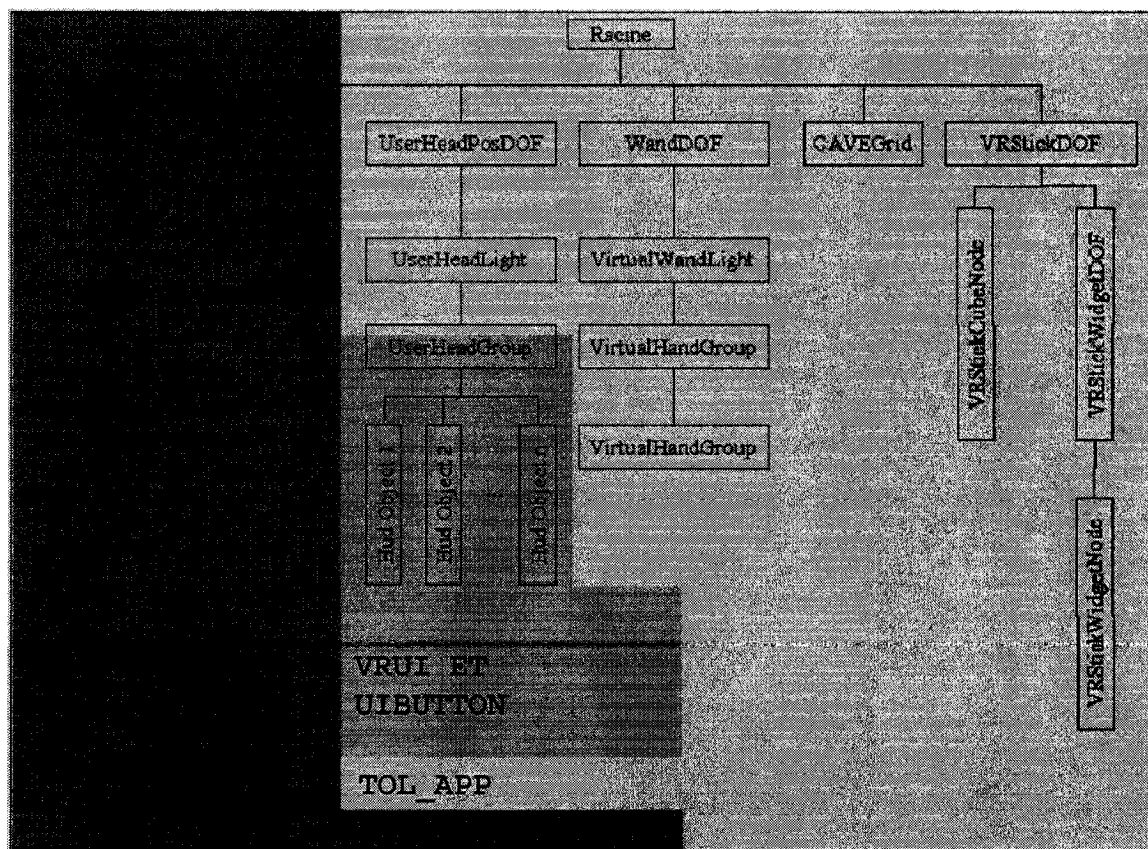


Figure 2.3. Description du graphe de scène utilisé

### 2.2.5 Structure de données : aperçu du graphe de scène

La Fig. 2.3 représente la structure de données utilisée pour gérer l'information. On distingue cinq branches sous la racine. C'est dans le premier embranchement que l'on retrouvera toutes les géométries de produits et de pièces mécaniques. Le nœud `UserWorldPosDOF` est une matrice de transformation servant à positionner le monde virtuel à l'intérieur de la CAVE. C'est cette matrice qui permet la navigation dans l'environnement virtuel. Les nœuds enfants de `UserWorldPosDOF` sont des transformations servant à positionner les éléments par rapport au monde. Leurs enfants sont les nœuds de géométrie « `ObjectNode` », et ces derniers peuvent eux aussi avoir des nœuds enfants afin de positionner d'autres objets par rapport à eux-mêmes. La première branche est donc gérée par la classe `Sim_Logic`.

Les quatre branches suivantes permettent de contrôler respectivement la position des menus flottants, le curseur de la souris 3D (la main virtuelle), un affichage de quadrillage statique qui coïncide avec les murs de la CAVE et finalement un outil de navigation pour se déplacer dans le monde virtuel. Ces branches sont donc principalement gouvernées par la classe `Tol_App` à l'exception des menus et de leur position relative à l'utilisateur qui sont contrôlés par les classes des menus `VRUI` et `UIButton`.

## 2.3 L'interface utilisateur

Selon l'approche proposée, l'analyse du tolérancement s'effectue selon le scénario suivant :

1. chargement des pièces en fichier STL;
2. définition des faces;
3. définition des tolérances sur chaque pièce;
4. définition des contraintes d'assemblages;
5. analyse - génération des volumes d'existence.

L'interface utilisateur offre la hiérarchie de menus tel qu'illustrée à la Fig. 2.4. Le menu principal illustré à la Fig. 2.5 permet d'accéder aux commandes qui permettent de manipuler les fichiers, les faces, les contraintes d'assemblage ou les outils d'analyse.

Fichiers	Surfaces	Tolérances	Assemblage	Analyse
Import d'un fichier	Définir une surface plane	Géométrique	Face-face	Collision
STL	Définir un cylindre	Dimensionnelle	Axe-axe	Distance min
Modèle de pièce	Effacer une surface/cylindre	Effacer		Distance max
Produit				Visualiser le volume
Sauvegarde				Mise à jour de l'analyse
D'une pièce				
D'un produit (scène)				
Effacer une pièce ou un produit				

Figure 2.4. Hiérarchie des menus

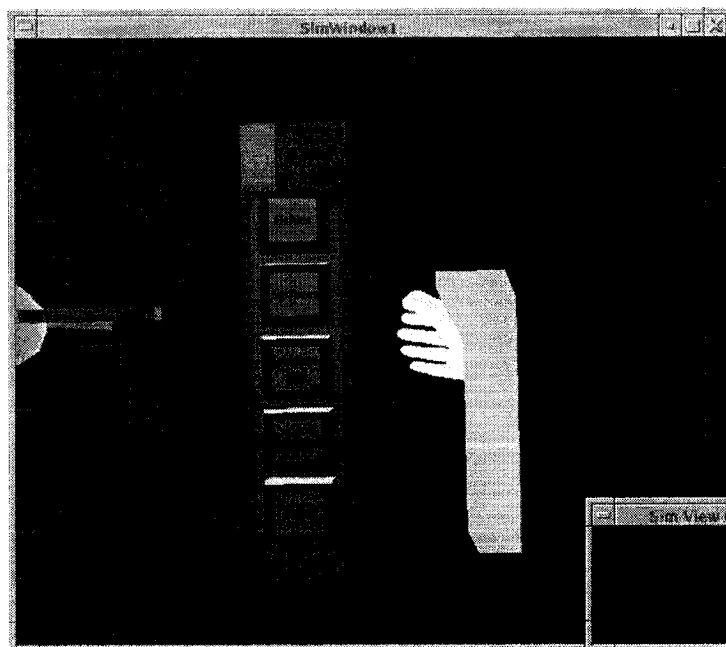


Figure 2.5. Aperçu du menu principal en mode simulation

La première étape consiste à charger les pièces. Le sous-menu de « fichier », illustré à la Fig. 2.6, offre la possibilité de choisir parmi les fichiers qui se trouvent dans un répertoire prédéfini. Par exemple, l'utilisateur qui clique sur STL verra le menu de la Fig. 2.7 afin de dérouler la liste des fichiers STL pour en choisir un à charger dans l'environnement de réalité virtuelle.

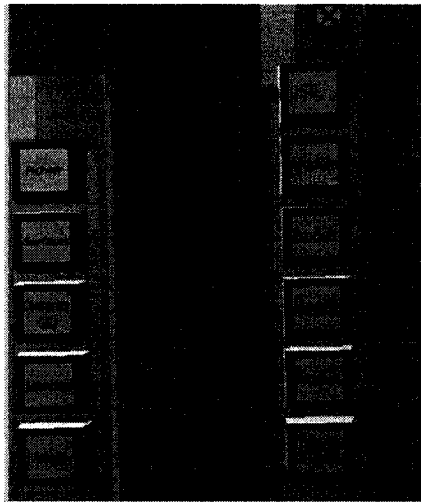


Figure 2.6. Aperçu du sous-menu de fichier

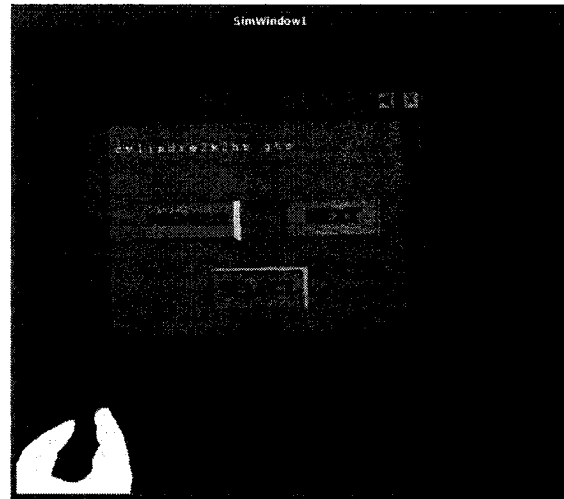


Figure 2.7. Interface usager pour la sélection d'un modèle à charger

Une fois les pièces chargées, il est possible de les manipuler avec la main virtuelle et les positionner dans la scène. Les surfaces fonctionnelles d'une pièce sont formées d'une ou plusieurs facettes du modèle STL. Il est nécessaire de regrouper les facettes appartenant à une même face avant d'appliquer les tolérances. D'abord il est nécessaire d'identifier les facettes de la pièce qui représentent une surface fonctionnelle. Cela est possible à partir du sous-menu « surfaces », illustré à la Fig. 2.8. Ces algorithmes ont été développés afin d'identifier les ensembles de facettes du modèle STL qui forment soit une surface plane ou une surface cylindrique. Donc à la Fig.2.8, après avoir choisi le type de surface à définir, il faut créer une boîte 3D qui inclut les facettes de la surface fonctionnelle. Un algorithme filtre les facettes en gardant celles qui ont toute la même orientation pour les faces planes. Dans le cas des surfaces cylindriques, l'algorithme

détermine quelles facettes sont susceptibles d'appartenir à une même surface cylindrique. Pour que cela fonctionne correctement, la majorité des facettes incluses dans la boîte de sélection doivent appartenir à la surface cylindrique. Un premier tri est fait en vérifiant que le produit vectoriel de la normale de deux facettes correspond à la direction de l'axe du cylindre. Lorsque l'axe du cylindre est déterminé avec une certitude acceptable, tous les vertex des facettes restantes sont projetés dans le plan normal à l'axe et le meilleur rayon du cylindre et le centre du cylindre sont déterminés. Les figures 2.9 et 2.10 montrent la boîte de sélection et la surface qui a été définie.

Lorsque les surfaces ont été définies, il est possible de les sélectionner afin d'appliquer des tolérances géométriques ou dimensionnelles ou des liaisons d'assemblage. La Fig. 2.11 montre l'interface qui permet d'appliquer les tolérances géométriques. La Fig. 2.12 illustre les quatre types de liaison d'assemblage qui ont été définies dans le cadre de ce projet: plan-plan avec jeu, plan-plan sans jeu, cylindre-cylindre avec jeu et axe-axe. Cette liste n'est pas exhaustive. Une liste plus détaillée des différents types de liaisons d'assemblage est disponible dans [11] et [13].

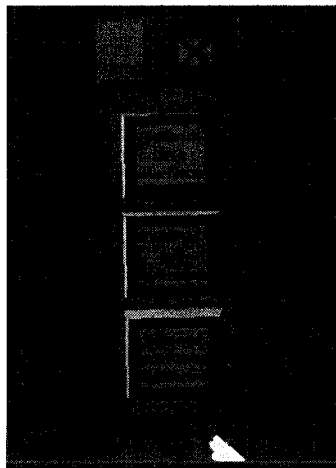


Figure 2.8. Aperçu du sous-menu de surfaces

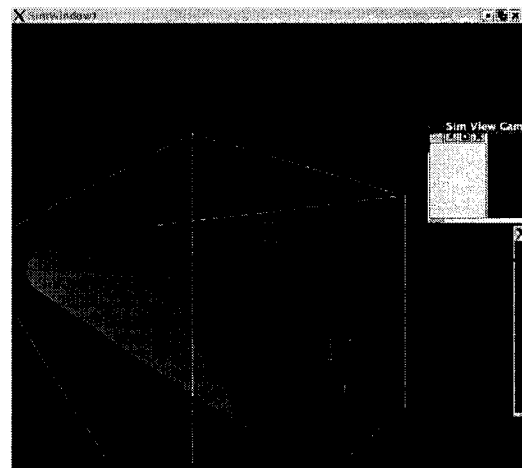


Figure 2.9. Sélection d'une face cylindrique en utilisant la boîte de sélection 3D

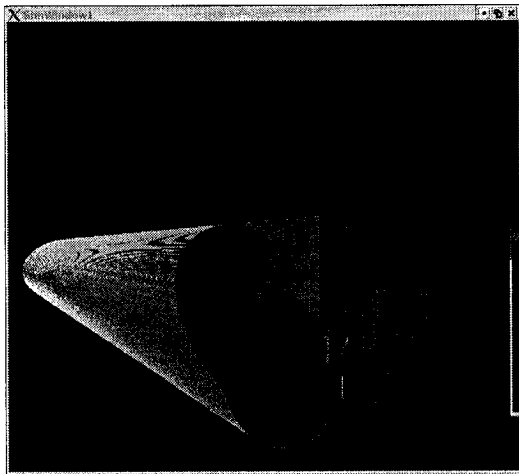


Figure 2.10. Résultat de la sélection d'une face cylindrique

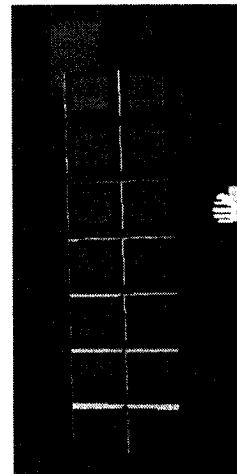


Figure 2.11. Aperçu du sous-menu des tolérances géométriques

La Fig. 2.13 illustre un assemblage de 3 prismes empilés les uns sur les autres. Les trois prismes sont insérés à l'intérieur d'une pièce en forme de « c ». Dans cet exemple, chaque prisme a une hauteur de  $2 \pm 0.25$  mètres et chaque coté mesure 1 mètre.

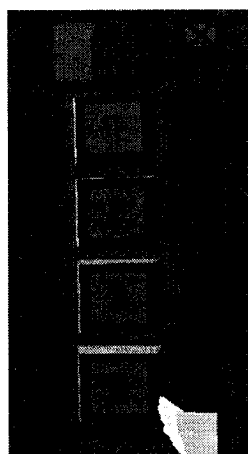


Figure 2.12. Aperçu du sous-menu des liaisons d'assemblages

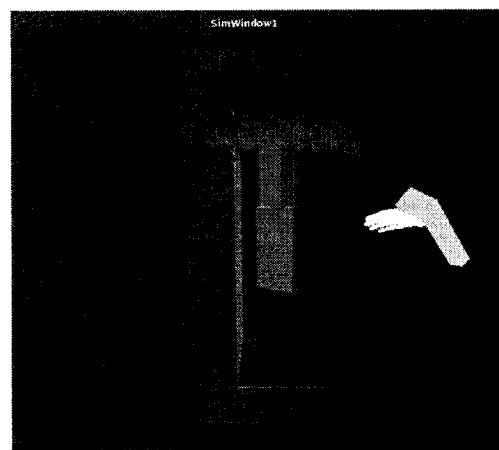


Figure 2.13. Volume d'existence d'une zone de tolérance



## 2.4 Analyse des tolérances

Le volume d'existence possible d'une surface permet de voir la zone dans laquelle tous les cas possibles de cette surface peuvent exister selon le tolérancement appliqué (Fig. 2.13). Dans certains cas, l'utilisateur peut donc constater immédiatement s'il y a une configuration possible qui fera que le produit ne s'assemblera pas ou sera défectueux. Par contre, il sera nécessaire d'examiner chaque surface individuellement et de la contraindre à un des pires cas pour mieux comprendre où se trouve l'erreur. Les commandes prévues sont illustrées à la Fig. 2.14.

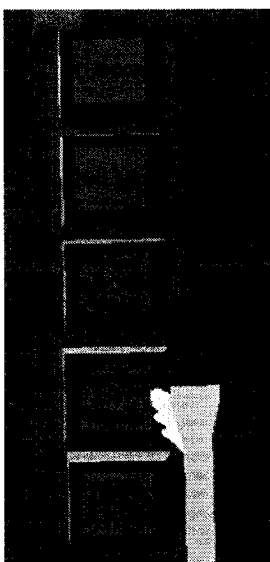


Figure 2.14. Aperçu du sous-menu d'analyse des tolérances

Pour l'instant, seule la fonctionnalité de visualisation du volume d'existence a été implémentée. Pour une pièce tolérancée qui ne fait pas parti d'un assemblage, le volume d'existence d'un élément topologique correspond à sa zone de tolérance. Comme on peut le voir la Fig. 2.15, le cylindre (d'une longueur de 2 unités) à une zone de tolérance sur une de ses faces (la zone fait 0.25 unité en longueur). À la Fig. 2.13, le volume en forme de parapluie représente le volume d'existence pour L'assemblage qui consiste de l'empilement de trois prismes.

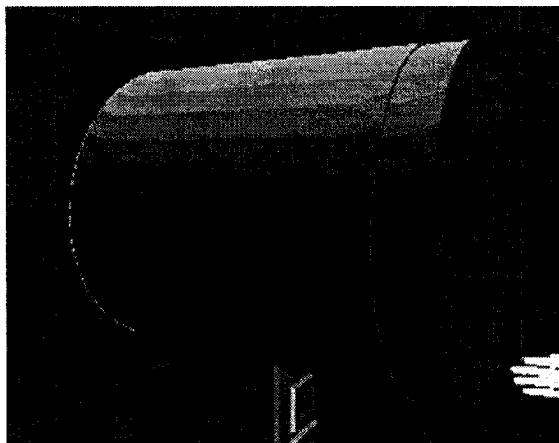


Figure 2.15. Pièce cylindrique et sa zone de tolérance

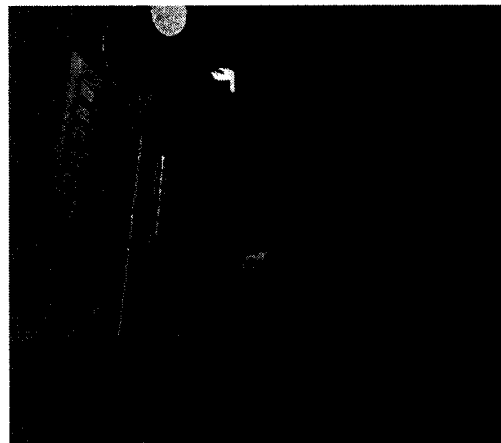


Figure 2.16. Manipulation de la zone de tolérance

Un volume d'existence est aussi un polygone instanciée séparément de la pièce et peut être manipulé et examiné indépendamment tel qu'illustré à la Fig. 2.16. La Fig. 2.17 montre le volume d'existence du troisième cylindre d'un empilement de cylindre. La pièce et son volume d'existence ont été déplacés par rapport au reste de l'assemblage.

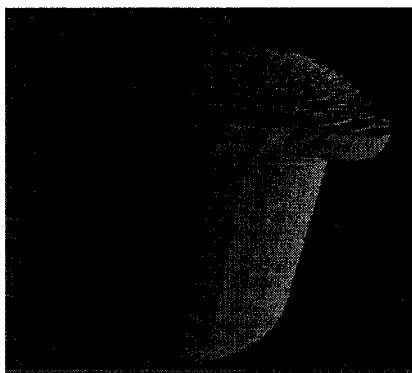


Figure 2.17. Volume d'existence d'une zone de tolérance

Pour obtenir le volume d'existence d'une face, la zone de tolérance de cette face doit être balayée sur les DDL qui existent sur le référentiel des faces liées par des liaisons d'assemblage dans un chemin (le concept de chemin est expliqué plus en détail au prochain chapitre). Le balayage de la zone de tolérance pour le troisième prisme

s'effectue d'abord à partir des DLL des faces impliquées dans la liaison plan à plan qui existe avec le second prisme. La zone de tolérance de la face supérieure du troisième prisme est balayée sur un DLL en translation et les deux DDL en rotation dans le plan du contact. Le résultat de cette opération est un volume d'existence intermédiaire. Celui-ci est à son tour balayé selon les DDL qui existent sur les autres liaisons qui suivent sur le chemin.

De façon générale, le volume d'existence d'une face est créé de la façon suivante :

- la face à analyser est sélectionnée ;
- son nœud correspondant est trouvé dans le graphe ;
- le volume de la zone de tolérance est généré ;
- tous les chemins entre ce nœud et un nœud fixe (élément de référence d'une pièce fixe) sont déterminés ;
- la zone de tolérance est balayé selon les DDL présents sur les éléments associés aux liaisons d'assemblage. Le résultat d'une opération de balayage est un volume d'existence. Le premier balayage d'un chemin est le balayage de la zone de tolérance de la face. Les balayages subséquents sont des balayages de volumes d'existence intermédiaires. L'ordre des balayages doit suivre l'ordre du chemin en commençant par ceux qui sont le plus proche de la face à analyser.

Par exemple, la Fig. 2.18 montre que pour l'assemblage de l'empilement de quatre blocs, le volume d'existence du bloc du dessus est déterminé en balayant d'abord la zone de tolérance autour des DDL à la liaison 1. Le volume résultant de cette opération ( $V_1$ ) est alors balayé autour des DDL de la liaison 2 et  $V_2$  est ensuite balayé autour des DDL de la liaison 3.

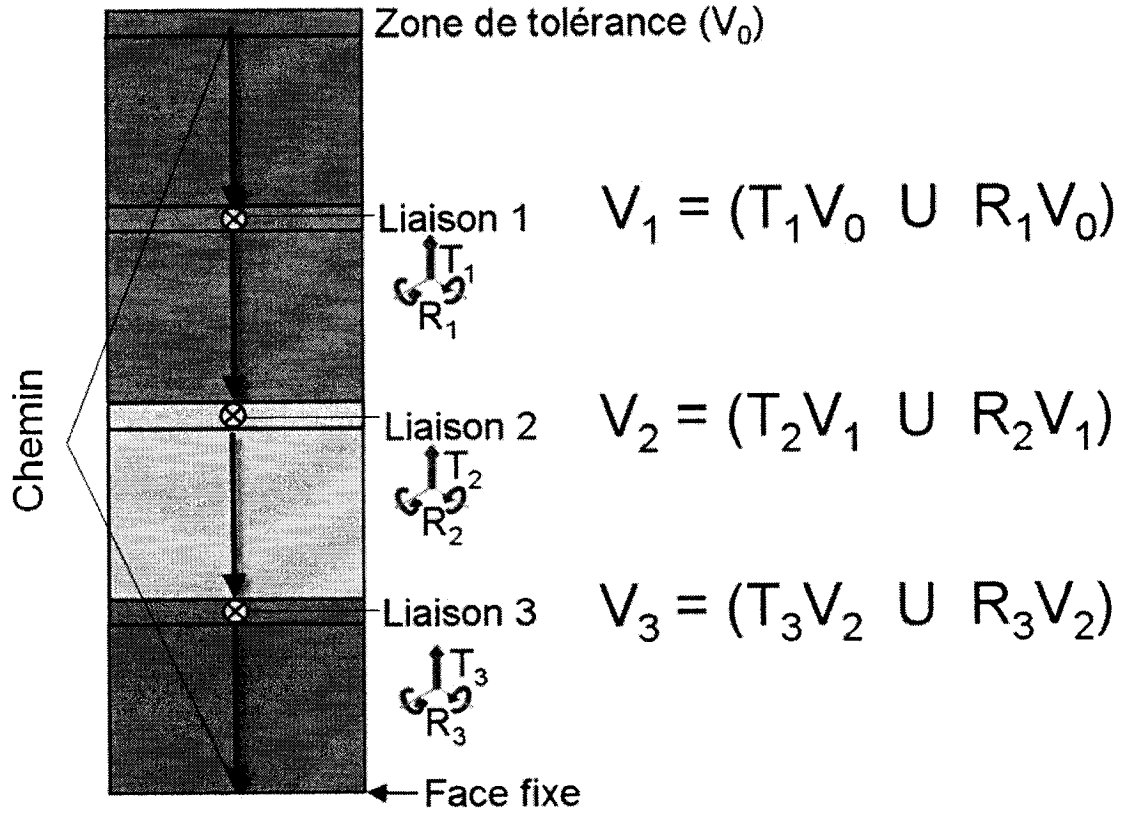


Figure 2.18. Procédure de génération du volume d'existence

Afin d'obtenir un volume d'existence de bonne qualité, quatre axes de rotation sont choisis pour effectuer le balayage des erreurs angulaires. La Fig. 2.19 montre une face de forme arbitraire avec les quatre axes de rotation qui croisent le référentiel de la pièce. Il y a 45 degrés entre chaque axe. La Fig. 2.20 montre l'angle de rotation maximal selon la largeur de la pièce suivant l'axe  $X_3$  et pour une tolérance de  $T$ . L'angle de rotation ( $\theta$ ) est limité par l'épaisseur de la zone de tolérance et la plus grande largeur de la face dans la direction. Dans cet exemple, l'angle de rotation maximal est calculé par :

$$\theta = \arcsin \left( \frac{\left( \frac{T}{2} \right)}{X_{3MAX}} \right) \quad (3.1)$$

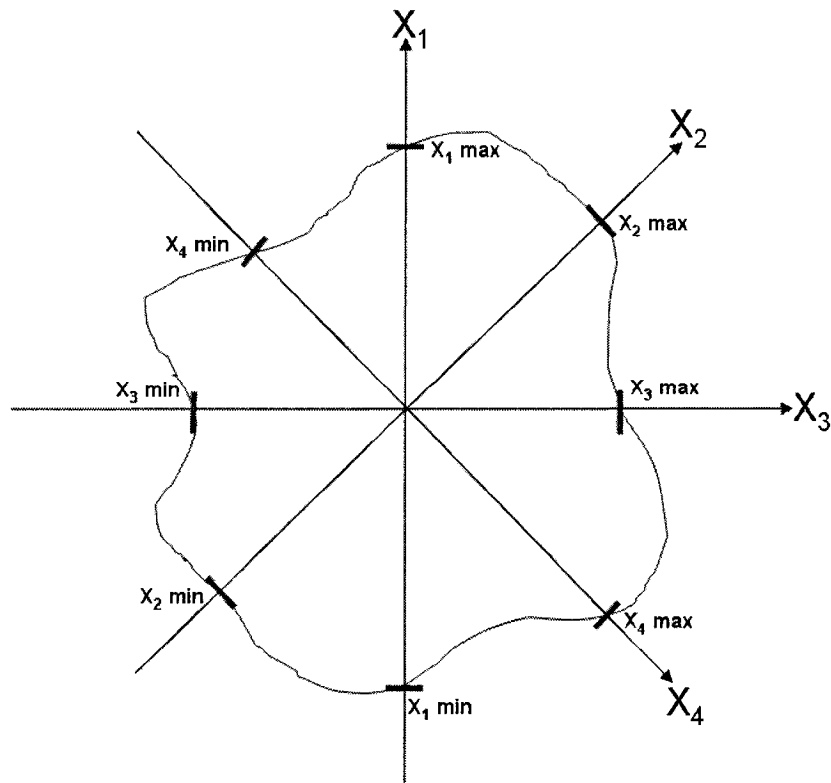


Figure 2.19. Les quatre directions de rotation

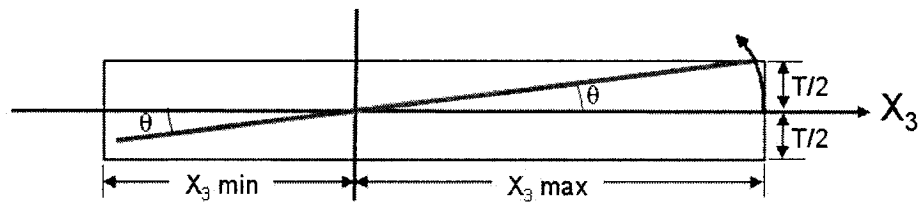


Figure 2.20. Angle de rotation maximal selon l'axe  $X_3$

L'algorithme crée deux matrices de transformation pour chaque direction angulaire et deux matrices pour les pires cas en translation. Une copie du volume d'existence est transformée selon ces transformations et une union booléenne combine les deux volumes. Pour les mécanisme à architecture parallèle, l'algorithme calcule d'abord les volumes d'existence pour chaque chemin indépendamment. Afin de tenir compte de

l'interdépendance des chemins, l'algorithme effectue ensuite l'intersection booléenne entre les résultats de chaque chemin.

Les commandes «collision», « min » et « max » n'ont pas été implémentées. L'objectif était de supporter une détection de collision afin de déterminer les risques de collision entre les pièces. Les commandes « min » et « max » auraient permis de calculer le jeu minimum et maximum en utilisant les volumes d'existences générés.

## **2.5 Situation des articles dans le mémoire**

Afin de pouvoir obtenir la représentation de l'assemblage et des éléments de variabilité telle qu'illustré à la Fig. 2.18, il est nécessaire d'automatiser l'interprétation des tolérances et obtenir les chaînes cinématiques qui bouclent la fonction de contrôle. En raison de plusieurs autres difficultés rencontrées lors du développement de la méthode graphique 3D, qui sont expliquées plus en détails au chapitre 5, et la nécessité d'automatiser l'interprétation des tolérances, le deuxième objectif de cette recherche a visé à développer cette fonctionnalité. Le premier article, qui forme le chapitre 3, propose une méthodologie afin d'obtenir automatiquement les boucles cinématiques à partir d'un graphe d'assemblage et aussi l'ensemble des tolérances qui peuvent affecter chaque élément de la chaîne cinématique. L'utilisation de graphe d'assemblage pour obtenir les chaînes cinématiques était déjà connue dans la littérature. Par contre l'utilisation du graphe de pièce utilisés dans le but d'obtenir les tolérances qui affectent un élément topologique est nouveau. Un exemple d'assemblage est proposé afin de comparer notre méthode d'interprétation des tolérances à une analyse 1D traditionnelle. Finalement, le deuxième article, qui compose le chapitre 4, explique les structures de données et les algorithmes informatiques qui ont permis d'implanter le processus d'automatisation des tolérances.

**CHAPITRE 3: ARTICLE I: METHODOLOGY FOR THE  
INTERPRETATION OF GEOMETRICAL AND DIMENSIONAL  
TOLERANCES IN AN ASSEMBLY**

Soumis à *International Journal of Advanced Manufacturing Technology*  
Édition *Springer*

Remy Dionne, Christian Mascle PhD, Luc Baron PhD

Department of Mechanical Engineering  
Ecole Polytechnique de Montreal  
C.P. 6079, succ. CV, Montreal, Quebec, Canada

Communicating author:

Christian Mascle

Email : [christian.mascle@polymtl.ca](mailto:christian.mascle@polymtl.ca)

Telephone: (514) 340-4711 ext. 4398

Fax: (514) 340-5264

### 3.1 Abstract

The analysis of geometrical and dimensional tolerances (GD&T) is composed of two steps. The first of these steps is the interpretation of the tolerances, which make it possible to obtain kinematic loops. The second step is the method of resolution. This article describes a new methodology for the interpretation of geometrical and dimensional tolerances in an assembly. Two graph data structures are introduced. The first graph, the part graph, represents the dimension and tolerances used at the part level. This data structure is used to store tolerance data in such a way as to facilitate the identification of the tolerance dependencies on any geometrical element. The second graph, the assembly graph, represents the various relationships between the parts of the assembly. The methodology relies on a procedure that generates the kinematics loops between two topological elements belonging to different parts of an assembly. It then identifies the affecting tolerances on the topological elements that are part of the loop. A use case is presented to illustrate each concept.

### 3.2 Key words

*Geometrical and dimensional tolerance, analysis, interpretation, ASME Y14.5M-1994, kinematic loops, assembly.*

### 3.3 Introduction

Being able to reduce the development time is necessary in order to meet the needs of consumers and to gain an advantage over the competitors. The average development cycle in the automotive industry has dropped drastically, from 60 months in the 1960s, to 18 months in the present. This can be explained in part by the improvement of design software and tools, as well as new methodologies which allow the re-use of the data models throughout the development process.

Currently on the market, there are many Computer-Aided Design (CAD) software packages that can be used to design and analyze parts and products. Despite advances in



the fields relating to the validation of a product before production, such as finite element analysis, ergonomic and kinematics analysis, and virtual reality, a significant gap still exists between the phases of design, industrialization and production.

This research is on a new approach for the analysis of Geometrical and Dimensional tolerances (GD&T). This stage of the development process establishes the link between the phases of design and industrialization. At the design level, it allows the designers to verify the interference between parts, optimize the assembly sequence, and ensure the functionality of the product. At the industrialization level, it allows for the optimization of the required machining quality and ensures the interchangeability of parts. GD&T analysis is often neglected and results in the necessity to rework parts, review the product design in order to bring corrective measures, and may require recourse to an alternative solution such as the use of shims.

The ultimate objective of tolerancing is to improve the quality of the product and to minimize the costs of operations. The main problem is that the current tolerance analysis solutions are not precise enough and the problem is very complex to solve. This explains why there is a need for further research to develop tolerance analysis software.

We divide the problem in two tasks: interpreting the tolerances and solving the tolerance analysis. The first task is to define a methodology to collect the significance of the tolerances before doing the analysis. The result is the set of kinematics loops and the tolerance dependencies for every topological element that is part of the loops. The second task is to solve the kinematics loops and analyze the results. This article reflects the work carried out to interpret the tolerances.

### **3.4 State of the Art**

#### **3.4.1 Commercial Software**

The problem with most tolerance analysis software is that they are designed to solve a specific problem. For example, spreadsheets are often used to calculate the dimension transfers necessary in order to determine machining operations. These software are often limited to uniaxial analysis.

With 3D CAD software, it is possible to apply tolerances to digital models. However, the purpose of doing such is strictly to dress up the models with 3D annotations. CAD software applications, such as CATIA and Unigraphics, ensure that the annotations conform to ISO or ASME standards. Tolerance analysis solutions are usually provided as add-ons from third party applications. CETOL Six Sigma and TASYSSWorks are two examples of such software, and will now be discussed in detail.

##### **3.4.1.1 TASYSSWorks**

TASYSSWorks is a tolerancing application integrated into SolidWorks. This system has three phases to apply, analyze and optimize tolerances. Each of these phases are executed separately by the applications PreTASYSSWorks, TASYSSWorks and TASYSSWorks Intol.

PreTASYSSWorks allows the user to define the control function. This is also sometimes referred to as a measurement for which the gap must be analyzed. First, the inner and outer bounds which should be respected must be defined. The software then seeks out all of the dimensions which have an influence on the control function. The user can then manually add missing or new dimensions. At this stage, it is possible to visualize the product when the control function is at its nominal, minimum, maximum or median value.

TASYSWorks makes it possible to study the effect of tolerances on the control function in a statistical or deterministic way. The statistical approach assumes a normal distribution for each tolerance. Therefore, it is possible to calculate the distribution of the control function which gives an indication on the fraction of parts that will be rejected. The deterministic analysis (“worst-case scenario”) calculates the combination of values that will result in the worst possible condition for the control function. Graphs indicate the contribution and sensitivity of each dimension on the error of the control function. Sensitivity is an indication of how much the control function changes when the tolerance is changed. The user then manually changes the tolerances by tightening tolerances that are sensitive and contribute more to the error and by loosening the tolerances that are not sensitive and contribute little.

TASYSWorks Intol is a module that automatically optimizes tolerances based on other factors, such as manufacturing costs. For example, the software can allocate tolerances in order to minimize the costs of manufacturing.

#### **3.4.1.2 Cetol Six Sigma**

Cetol Six Sigma is a plug-in for CATIA V5 or Pro/E. When the three-dimensional model is completed within CATIA, relevant information is shared with Cetol Six Sigma. Cetol Six Sigma then uses this information to create its own tolerance model. The analyses obtained are similar to those of TASYSWorks (contribution graph, sensitivity graph, statistical parameter CEK, analysis reports, etc).

Cetol Six Sigma is based on the work of Dr. Kennett W Chase and the ADCATS program at Brigham Young University, and is developed by Sigmetrix. The tolerance analysis is conducted through detecting a vector loop and constructing transformation matrices for small displacements. The reader may refer to [27,28] for more information about the method of resolution used.

### 3.4.2 Review of the Literature

The assemblability of a product is evaluated by verifying the clearance between mating surfaces. A negative clearance indicates interference. The control function indicates between which surfaces the gap must be evaluated. Three tasks are required to carry out a tolerance analysis:

- 1) Identification of the kinematic chains which imply the control function;
- 2) Interpretation of the tolerances;
- 3) Solving for the control function.

With the 1D traditional uniaxial analysis approach, the user has to study both the assembly and the parts, understand their operations in order to identify the kinematic chains implied in the control function, interpret the significance of the dimensions and tolerances, and then solve the problem. This process is prone to human error and is time consuming. The resolution is simple to solve, but it employs many simplifications.

In 1988, the International Workshop on Mechanical Tolerancing held in Orlando, Florida, identified the need for the mathematization of tolerances. A committee was founded shortly afterwards with the goal to create explicit mathematical definitions for the use of tolerances. This led to the creation of the ASME Y14.5.1M-1994 standard. The ASME Y14.5.1M-1994 standard [29] partly clarifies the interpretation of tolerances. The mathematical definitions for tolerances are defined in [30].

Two fields of research are of interest to us. First of which is research aimed at obtaining a logical representation of tolerances. This includes defining a data structure for the purpose of storing information about the tolerances and which represents the relations between the tolerances and the geometry. The second field of interest is establishing a mathematical method to model tolerances.

### **3.4.2.1 Tolerance Representation**

This research is partially inspired from the Technologically and Topologically Related Surfaces (TTRS) theory developed by André Clement and Alain Desrochers. In [11], the TTRS theory is used to generate the dimensions and tolerances. This is the reverse process that we wish to attain, but this technique could also be applied to tolerance analysis.

TTRS is a concept which uses graph theory to associate mating surfaces together in an assembly and to create an axis system on the parts. Each node of the graph represents a part, and the edges between them are assembly constraints.

Requicha developed the VGraph [17]. VGraph is a data structure which represents the features, surfaces, datums and tolerances of a part as a hierarchical graph. This concept differs from that of TTRS since the graph is adapted to represent existing dimensions, whereas the concept of TTRS is used to define dimensions. Moreover, the VGraph considers the part independently of its function in the assembly, whereas the TTRS requires the assembly. Finally, although Requicha's VGraph illustrates the association between tolerances and surfaces, it does not indicate a way of carrying out tolerance analysis.

Prabhu and Al. proposed an object oriented data structure to represent tolerances [31]. However, the utility was limited to an assistant for the application of the tolerances on features recognized by the software.

### **3.4.2.2 Mathematical Methods**

#### **4.4.2.2.1 The Parametric Model**

A parametric modeler allows a part to be built by using design constraints. The part can then be modified by changing the constraints. Tolerance analysis is performed by translating the tolerances into the design parameters.

Each parameter is attributed an amplitude of variation which is based on the tolerances. By varying each parameter, a population is generated which makes it possible to study different configurations of an assembly. A large number of configurations can be generated in order to carry out a statistical analysis. This technology was used often in combination with uniaxial analysis techniques. An alternative of variation analysis involves meshing the part and distributing the tolerances to the nodes. Some current 3D tolerance analysis software create an abstract model composed of geometric primitives defined by a set of constraints that derive from the interpretation of the tolerances.

#### **3.4.2.2.2 The Vectorial Model**

The vectorial model introduces a position and an orientation vector for each surface. Several authors have worked out techniques to replace tolerance zones by variation matrices. A variation matrix is the product of one or more homogeneous transformation matrices, which represent the permissible variations of the nominal surfaces about each degree of freedom. By inserting the variation matrices into the kinematic chain and by generating a population of results, it is possible to study tolerances at the control function. An alternative statistical approach consists in allotting a distribution to each variation, and then calculate what the distribution would be for the control function. Several alternatives of this approach exist. The equivalent can be accomplished with vectorial expressions rather than matrices.

#### **3.4.2.2.3 The Small Displacements Twists Model**

Desrochers, Laperriere and Ghie contributed to develop a method which uses small displacement twists [10,12,13]. Unlike vectorial approaches which must generally study a great number of configurations to obtain an indication of the tolerance zone, Desrochers represents the tolerance zone by variation twists. The kinematic chains and the tolerance zones are also represented by twists. A jacobian matrix is built to map variation twists to

the same reference frame as the control function. The effect of the tolerance zone on the control function is obtained by multiplying the jacobian matrix with the variation twists.

### 3.5 Basic Concepts

This section is a review of the basic concepts of tolerancing. The integration of each concept in the proposed method is elaborated in sections 3.6 and 3.7. For more information on tolerances, the reader should refer to [29].

#### 3.5.1 Definition of the G&D Tolerances

Technical drafting annotations can be either dimensional or geometrical. In order to model the tolerances, it is important to understand the properties that are specific to each category.

##### 3.5.1.1 Dimensional Tolerances

For dimensional annotations, the tolerance zone belongs to the dimension. This means that the tolerance zone is not defined specifically on one of the two elements linked by the annotation. Examples of dimensional tolerance annotations are shown in Fig. 4.1.

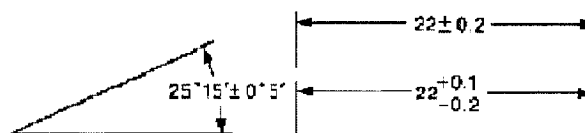


Figure 3.1. Examples of dimensional tolerances

##### 3.5.1.2 Geometrical Tolerances

Unlike dimensional tolerances where the tolerance zone belongs to the annotation, geometrical tolerances define the tolerance zone on a geometrical element. Also, a geometrical tolerance zone may be subject to the influence of reference geometry. There are 14 different kinds of geometrical tolerances, split into five categories: form, profiles, orientation, location and runout. See Fig. 3.2.

	TYPE OF TOLERANCE	CHARACTERISTIC	SYMBOL	SEE:
FOR INDIVIDUAL FEATURES	FORM	STRAIGHTNESS		6.4.1
		FLATNESS		6.4.2
		CIRCULARITY (ROUNDNESS)		6.4.3
		CYLINDRICITY		6.4.4
FOR INDIVIDUAL OR RELATED FEATURES	PROFILE	PROFILE OF A LINE		6.5.2 (b)
		PROFILE OF A SURFACE		6.5.2 (a)
FOR RELATED FEATURES	ORIENTATION	ANGULARITY		6.6.2
		PERPENDICULARITY		6.6.4
		PARALLELISM		6.6.3
	LOCATION	POSITION		5.2
		CONCENTRICITY		5.11.3
		SYMMETRY		5.13
	RUNOUT	CIRCULAR RUNOUT		6.7.1.2.1
		TOTAL RUNOUT		6.7.1.2.2
* ARROWHEADS MAY BE FILLED OR NOT FILLED				3.3.1

Figure 3.2. Summary of the geometrical tolerances (ASME Y14.5M-1994 [29])

#### 3.5.1.2.1 Tolerances of Form

The tolerance zone can correspond to a cylinder, the space between two concentric circles, two planes or two parallel lines. Tolerances of form control the quality of an individual surface independently of others and they are neither positioned nor oriented in space.

#### 3.5.1.2.2 Tolerances of Profile

The tolerance zone of profile corresponds to space between two parallel curves or two surfaces. They can be independent or related. If they are related, the position and the orientation of the tolerance zones are dependent on the associated datums.

#### 3.5.1.2.3 Tolerances of Orientation

The tolerance of orientation defines a zone located between two parallel planes whose orientation is defined relative to a reference datum. The orientation of the tolerance zone is defined but the position is not.



#### **3.5.1.2.4 Tolerances of Location**

Nominal annotations are used to indicate the exact position of a feature, and a positioning tolerance is added to indicate the acceptable deviation error from nominal. Whereas with a dimensional tolerance the tolerance zone belongs to the dimension, with a tolerance of position the dimension is exact and the tolerance zone is on the feature.

#### **3.5.1.2.5 Runout Tolerance**

The runout tolerance controls the form and orientation of a surface of revolution. There are two types of runout: circular and total. A circular runout controls the circularity, size, and variation of form for a section of a surface of revolution. A total runout controls the same as a circular runout, with the addition of controlling cylindricity and concentricity, since the entire length of the surface is controlled. The datum is an axis of revolution, which may be composed of two datums.

#### **3.5.2 Precedence of the Tolerances**

Each type of tolerance has its own specialty and use. They can control the form, orientation, position, dimension or the size of a feature. For example, a tolerance of orientation also implies a control of the form, but a tolerance of form does not imply a control of the orientation. Also, a position tolerance implies a control of the orientation but a tolerance of orientation does not control the position. Depending on the product's functional requirements, it may be necessary to control all of these aspects with different degrees of accuracy. On Fig. 3.3, which gives an example of the positioning of a hole, the hole must be positioned with precision in order to ensure a correct assembly with its counterpart. However, even with good positioning, the hole could have an angle which would violate the required perpendicularity. Thus, a perpendicularity tolerance is also required. Finally, it might also be necessary to control the straightness of the hole's axis.

This example shows that there is an order to follow when applying tolerances and that it is necessary to choose good constraints to satisfy the functional analysis. The position

tolerance is larger, and thus creates a zone in which the smaller tolerance zone of orientation can move. The tolerance zone of form is the smallest, and it is free to move inside the tolerance zone of orientation, as illustrated by the arrows of Fig. 3.4.

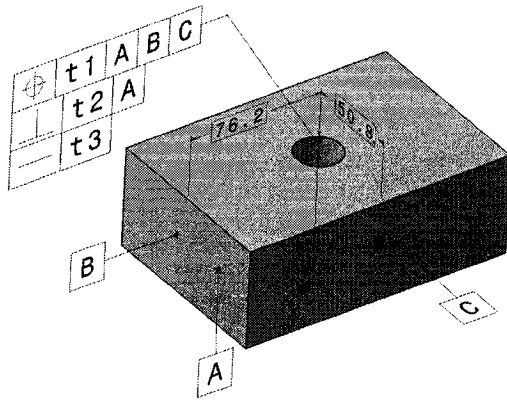


Figure 3.3. Controlling the positioning, orientation and form of a hole's axis

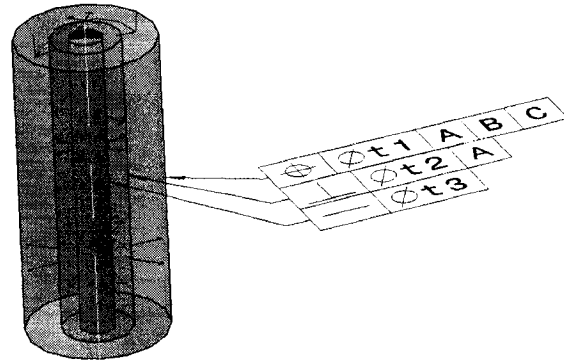


Figure 3.4. Representation of the three tolerance zones of the hole's axis

### 3.5.3 Datums

Dependent tolerances are positioned or oriented with respect to one or more references. These references, or datums, are one of three types of geometric elements: a planar surface, an axis, or a middle plane. Datum planes and axis constrain different degrees of freedom (DOF).

A plane datum constrains the translation normal to that plane and the rotation around the two axes in that plane. The axis datum constrains 4 DOF: the two translations and the two rotations normal to that axis.

In addition, if there is a tolerance zone on the datum, the datum's error will propagate to the toleranced surfaces along the DOFs that are constrained. In Fig. 3.5, the dotted arrows indicate the DOF which are constrained for each type of datum.

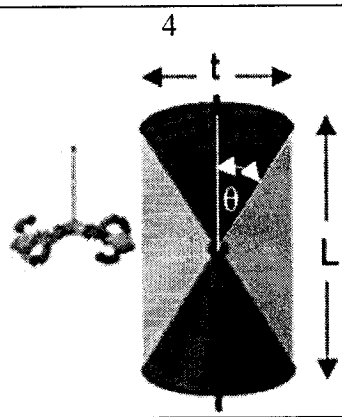
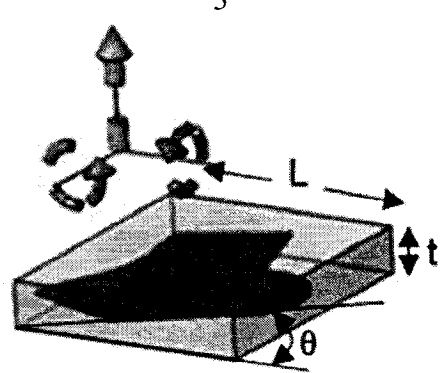
Type of datum	Tolerance zone	Constrained DOF
Axis	Cylindrical	<p>4</p> 
Plane	Prismatic	<p>3</p> 

Figure 3.5. Types of tolerance zones and the degrees of freedom they constrain.

#### 3.5.4 Worst cases

A statistical analysis consists of allotting to the tolerances a normal distribution on each DOF and to check what the distribution obtained will be on the control function.

Conversely, for a deterministic analysis, it is important to define the extreme cases. For a uniaxial tolerance analysis, there are two extreme cases for each tolerance: the Least Material Condition (LMC) and the Maximum Material Condition (MMC), or, for dimensional annotations, respectively the largest and the smallest dimension.

In order to carry out a deterministic analysis in 3D, it is necessary to develop the concept of worst angular cases about each DOF. The worst angular case must be considered

under two circumstances: when there is a stacking of tolerances, and for the mating surfaces of an assembly.

When there is a tolerance stack-up, the tolerance zone is swept about each DOF that is constrained by the primary, secondary and tertiary datums. Fig. 3.5 illustrates the amplitude of movement for a planar and an axis datum. The angular amplitude of movement ( $\theta$ ) around each DOF corresponds to  $\theta = \text{atan}(t/L)$

In Fig. 3.5, the volume of the axis swept about its DOF (without any translation) corresponds to two opposite cones.

It is slightly different with functional surfaces because only the area of support is considered. The assumption is that the surface of support will be the largest. On the profile view of a planar surface, the profile can be rising, leveled or descending. The extreme angular case is 1/3 rising followed by 1/3 leveled and 1/3 descending. The worst possible slope for the functional surface is  $\theta = \text{atan}(3t/L)$ . We are suggesting this hypothesis for the worst case tolerance analysis in 3D.

### **3.5.5 Rules for Constructing the Reference Frames**

There can be up to 3 datums on a tolerance. Several approaches propose to create a reference frame with the datums provided from which the tolerance zone can vary. Standard ASME Y14.5.1M-1994 proposes such an approach, but admits that the idea was the object of much debate. In his research to provide a kinematic formulation for tolerances, Rivest [32-33] presents the construction process for the reference frames that are used in his examples. The constraining of the DOF of a tolerated element depends on the type and order of its datums. This information must be captured by the interpretation of the tolerances for use during analysis.

### **3.6 Proposed Approach**

These are the steps of the tolerance interpretation process:

- 0) Modeling the parts with the nominal dimensions
- 1) Identifying the topological elements
- 2) Defining the GD&T annotations
- 3) Building the part graphs
- 4) Defining the assembly connections
- 5) Building the assembly graph
- 6) Identifying the kinematic loops

The interpretation of the tolerances process begins when a model is completed with its nominal dimensions (step 0). The dimensions and tolerances link with the topological level of the model (vertices, edges, faces, axis, etc). Each topological element is identified (step 1) and then the GD&T annotations are applied (step 2). The annotations are used to build one part graph for each part of the assembly (step 3). Then, assembly joints are defined between the parts (step 4) and the assembly graph is created (step 5). Control functions are defined and the kinematic loops around them are obtained from the assembly graph (step 6). The tolerance dependencies can be determined for each element of the kinematic loop from the part graphs.

An interesting aspect of this approach is that it can be used independently from the analysis method. In this article, the interpretation of the tolerance methodology is applied with a uniaxial analysis method. This puts in perspective the simplifications that occurs when doing a uniaxial analysis, and demonstrates that the interpretation method contains additional information that can be used in a 3D analysis.

### **3.6.1 Creation of a Part Graph**

The part graph is used to determine the dependencies of a surface. It is a dependency graph meaning that it is directional and should not contain cycles. Each node of the graph represents a topological element and each edge represents a GD&T annotation.

The orientation of an edge that represents a geometric tolerance is oriented. The orientation of an edge that represents a dimensional tolerance is ambiguous because the tolerance zone belongs to the dimension rather than to one of the linked surfaces. A direction is determined when analyzing how they are situated relative to the nodes that represent datum elements. Therefore, edges that represent dimensional tolerances are created as bi-directional because the tolerance zone can be on either the starting or the target node and the decision must be made at the time of analysis. An independent tolerance is represented by an edge that starts and ends on the same node.

Tolerances that have more than one datum are represented by an equal number of edges. An index on the edge indicates the priority of the datum (primary datum = 1, secondary datum = 2, tertiary datum = 3). Knowing the type of each datum and the edge index can be used during the analysis to determine the degrees of freedom that are constrained by the datum. A tolerance which has only one datum always has an index of 1.

When the part graph is completed, it is possible to find the dependencies for any topological element by building the spanning tree of a node. The spanning tree is obtained when the graph is transformed into a tree whose root node is the element in question. For this project, the programming was carried out in C++, and the BoostGraph library was used for building and manipulating the graphs.

Fig. 3.6 presents an example part with features positioned relative to datum surfaces that have form and orientation tolerances applied on them (example taken from [34]). Fig. 3.7 shows how the topological elements were identified. Fig. 3.8 shows the graph of the part.

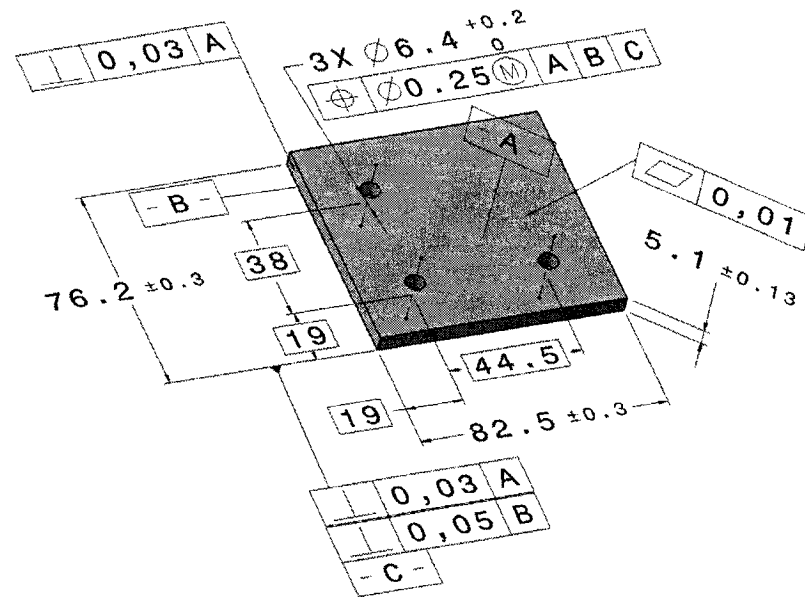


Figure 3.6. Geometrical and dimensional annotations on a part

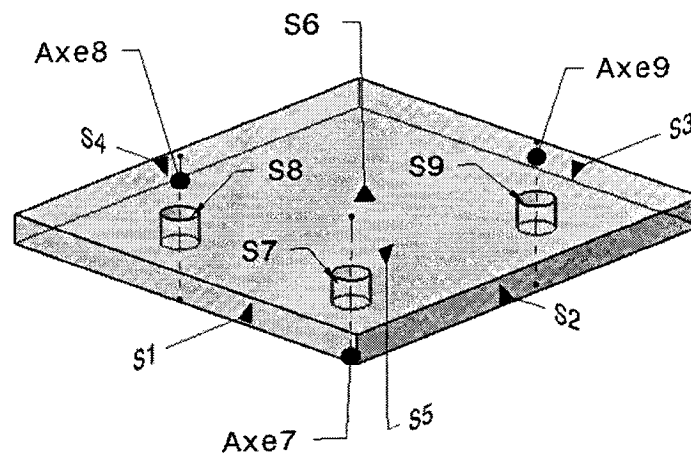


Figure 3.7. Identification of the topological elements of the part

The circles represent the nodes of the graph. The text inside the circles indicates the geometric element with which it is associated. Underlined text inside the node is used to indicate that the node is also used as a datum. The edges between two nodes represent a tolerance. The tolerance symbol is on the edge and the tolerance information is indicated nearby. Two other symbols are used to represent the dimensional tolerance and tolerance of size on a cylindrical geometry. The arrows on edges always point towards the





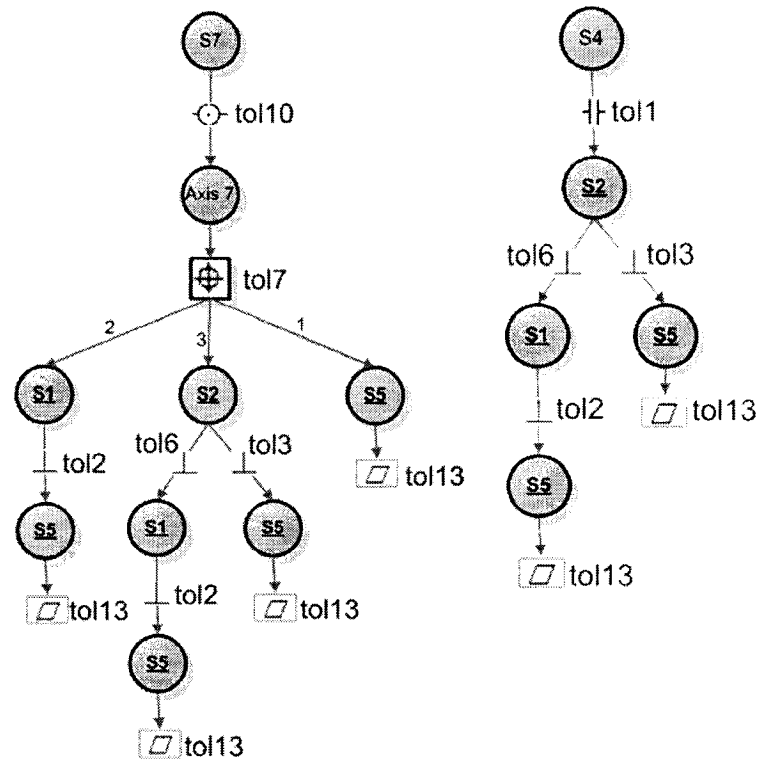


Figure 3.9. The dependency tree for surfaces S7 and S4

### 3.6.2 Creation of an Assembly Graph

In the assembly graph, each node represents a part. The edges correspond to assembly constraints between the parts. Each assembly constraint models the relative movement allowed between two geometric elements on different parts. Three types of assembly constraints, shown in Fig. 3.10, will be defined in this article and used in the example. Similar reasoning can be applied to define other assembly joints.









Constraint	Symbol	Free DOF	Constrained DOF
Axis to axis		None	
Plane to plane			
Cylinder to cylinder			

Figure 3.10. Types of assembly constraints, their symbol and affected DOFs

I) Axis to axis constraint: The axes of two parts are perfectly aligned. This is the hypothesis that will be made between the screw and its threaded hole. All the DOF are blocked and the two parts act as if they were only one. In reality, the screw and the threaded holes both have tolerances, but this simplification is acceptable considering that the threaded hole projects its tolerance onto the screw.

II) Cylinder to cylinder constraint: Is used if a peg enters in a hole of larger diameter. The link is made between the two cylindrical surfaces rather than between the axes. The peg is free to move in the axial direction but is limited in its translation and rotation movement about directions that are normal to the axis of the hole. The amplitude of movement depends on the diameters and the length of the hole and the peg.

III) Plane to plane constraint: The normals of two surfaces are aligned and the faces make contact. However, the parts are free to move in translation in the plane and in rotation around the plane's normal.

The assembly graph makes it possible to generate the kinematic loops that are related to the control function. A kinematic loop is the sequence of topological elements and assembly connections that bridge the two inputs of the control function. For example, in

the part graph of Fig. 3.11, the only kinematic loop for the control function that is defined between S1 and F2 is the following path: From S1 to S2, through the axis to axis constraint, and from F1 to F2. In this example, there is only one path between S1 and F2, since there is only one cycle, but as will be seen in the next example, a control function can have more than one path.

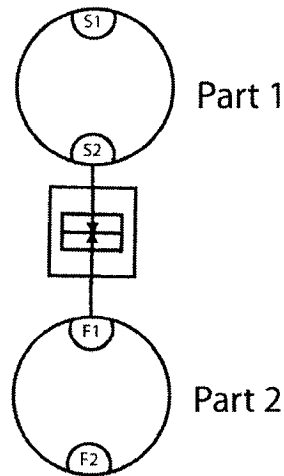


Figure 3.11. Example of an assembly graph

The kinematic loops are determined automatically using a function that is similar to the one used to generate the spanning trees as described in section 3.6.1. The difference is that since the assembly graph can contain cycles, it is necessary to take into account the fact that a node can only be used once in a kinematic loop. Therefore, the function described in section 3.6.1 is given a starting node as a parameter and the result obtained is the spanning tree for that node. The branches of the spanning tree that do not contain the end node are filtered out. The result is the set of kinematic loops that are associated to the control function.

### 3.7 Implementation

A rail assembly is used to demonstrate the suggested methodology. Primarily because this type of assembly is common in the industry, and despite the appearances, it is a complex assembly: It combines the three types of assembly constraints that were

presented in section 3.6.2, the parts are defined using various GD&T annotations, and the control function will consist of several kinematic loops.

The rail assembly illustrated in Fig. 3.12 contains four parts: the rail, the block and two screws. The detail drawings (Fig. 3.13 and Fig. 3.14) show that there are dimensional, form, orientation and positioning tolerances applied. The assembly constraints are plane-to-plane between the block and the rail, axis-to-axis between the rail and the screws, and cylinder-to-cylinder between the screws and the block.

The spacing between the rail and the block will be calculated in order to evaluate the assemblability of the product. The tolerance analysis problem will be solved in two ways. The first is by using the traditional uniaxial method, and the second is by using the suggested method for tolerance interpretation coupled to a modified uniaxial method. The comparison of the two methods will highlight the feasibility, functionality and the advantages of the suggested approach. Moreover, emphasis will be placed on the possibility and advantages of solving the problem in 3D.

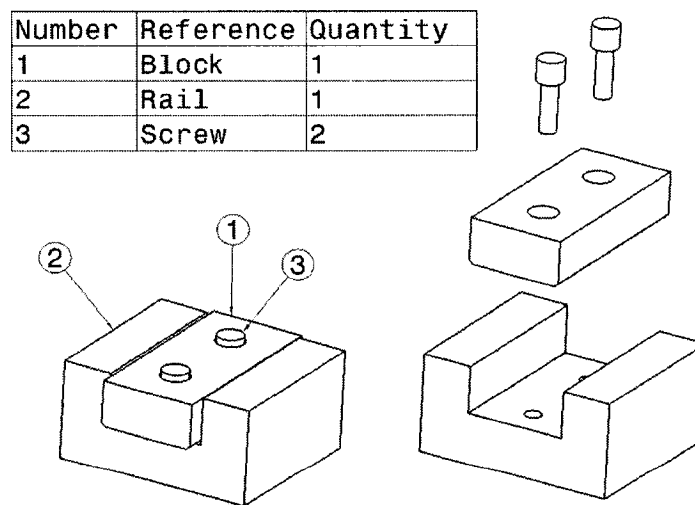


Figure 3.12. The block on rail assembly

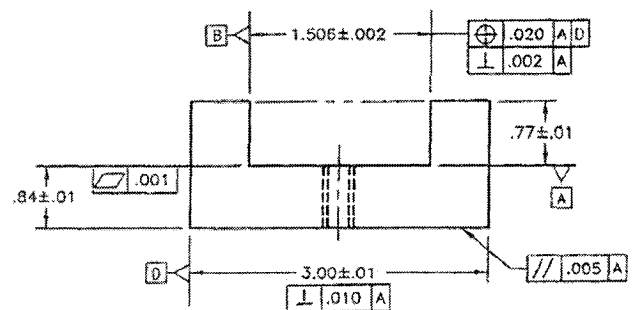
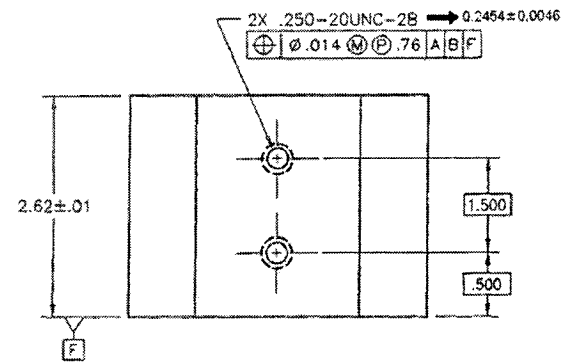
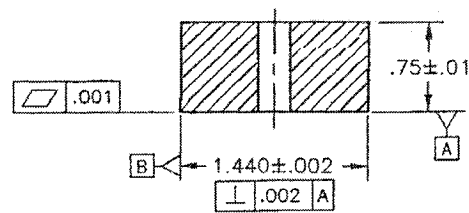
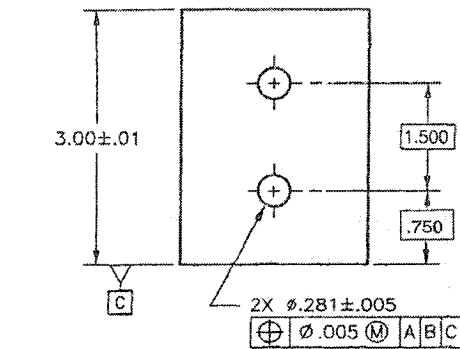


Figure 3.13. Detail drawing of the block

Figure 3.14. Detail drawing of the rail

### 3.7.1 Resolution by the Traditional Uniaxial Method

The 1D tolerance analysis of this problem follows the method in chapter 7 of [4]. The key elements of the solution are covered in order to emphasize the difference between the traditional approach and the proposed approach.

#### 3.7.1.1 Virtual Condition

The following virtual conditions will be used in the analysis:

- $R_G$  : width of the virtual condition of the slot on the rail:  $R0,753 \pm 0,002$
- $G_L$  : width of the virtual condition of the block:  $R0,720 \pm 0,002$
- $V$  : diameter of the virtual condition of the screw:  $R0,1227 \pm 0,0093$
- $G_T$  : virtual condition of the hole on the block:  $R0,1430 \pm 0,0075$

### 3.7.1.2 Explanation of the Virtual Conditions

#### **The rail's slot:**

The slot of the rail has a dimension of  $1,506 \pm 0,002$  a perpendicularity tolerance of 0,002 inches, and a positioning tolerance of 0,020. However, only the perpendicularity tolerance has an effect on the way the block is assembled with the slot in the rail. The position tolerance has no effect in this analysis because the holes are positioned relative to the slot. The virtual condition therefore has a width of  $1,506 \pm 0,004$ , or  $R0,753 \pm 0,002$  when expressed as a radius.

#### **The block's width:**

The block has a size of  $1.440 \pm 0,002$  and a tolerance of perpendicularity (0,002). Therefore, the virtual condition expressed as a radius is  $R0,720 \pm 0.002$ .

#### **The diameter of the screw on the rail:**

The screw and the rail can be regarded as being only one element because the hole projects its tolerance on the screw. The axis to axis constraint binds them together. In this case, the screw inherits the position tolerance of the hole (0,014). According to the Machinery Handbook, the diameter of the screw mounted in the hole is  $0,2454 \pm 0,0046$  (the minimum inner diameter of screw is 0,2408 and major diameter of the hole is 0,2500). Taking into account the position tolerance, the virtual state of the screw is  $0,2454 \pm 0,0186$  ( $R0,1227 \pm 0,0093$ ).

#### **The holes in the block:**

The holes have a diameter of  $0,281 \pm 0,005$  and a position tolerance (0,005 MMC). Taking into account the MMC modifier, the virtual state of the hole is  $0,286 \pm 0,015$  ( $R0,1430 \pm 0,0075$ ).

### 3.7.1.3 Effect of the Assembly

The assembly configuration in which the block is closest to the rail will be analyzed first, followed by the configuration which results in the greatest spacing between the two. For each configuration, the minimum and maximum gap will be calculated.

The minimum gap configuration occurs when the block is in contact with the screw as illustrated in Fig. 3.15. The kinematic chain begins on the face of the slot on the rail. It then passes by the rail's hole axis, which also corresponds to the screw's axis. Lastly, it goes to the cylindrical surface of the screw in contact with the inner wall of the block's hole, and then to the side face on the block. The path taken depends on the dimensions and tolerances, and must always pass by coincident geometries to cross from one part to another. The loop's solution is indicated below.

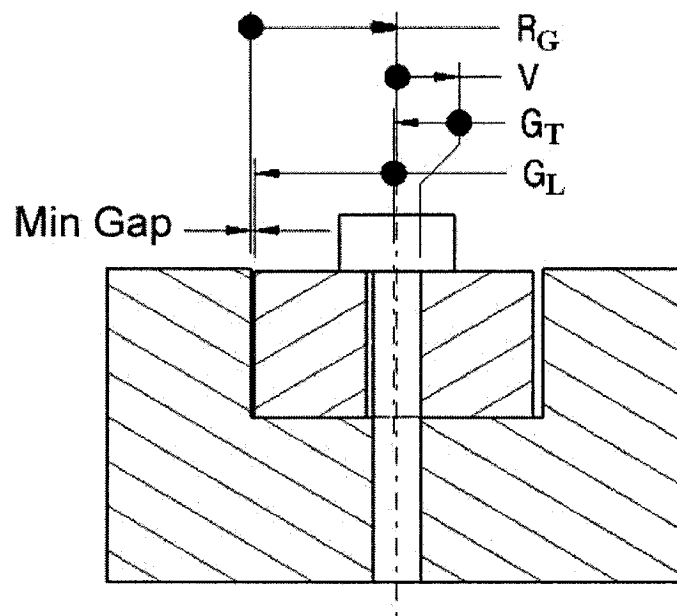


Figure 3.15. Minimum gap configuration for the analysis

$$\text{Average\_gap} = R_G + V - G_T - G_L \quad (3.1)$$

$$\text{Average\_gap} = 0,753 + 0,1227 - 0,1430 - 0,720 = 0,0127 \quad (3.2)$$

$$\text{Sum\_tol} = 0,002 + 0,0093 + 0,0075 + 0,002 = 0,0208 \quad (3.3)$$

$$\text{Min\_gap} = 0,0127 - 0,0208 = -0,0081 \quad (3.4)$$

$$\text{Max\_gap} = 0,0127 + 0,0208 = 0,0335 \quad (3.5)$$

The configuration which results in the maximum gap occurs when the block is in contact with the screw on the opposite wall as illustrated in Fig. 3.16. The kinematic loop is the same as for the min gap configuration, except that the orientation of vectors  $V$  and  $G_T$  is reversed.

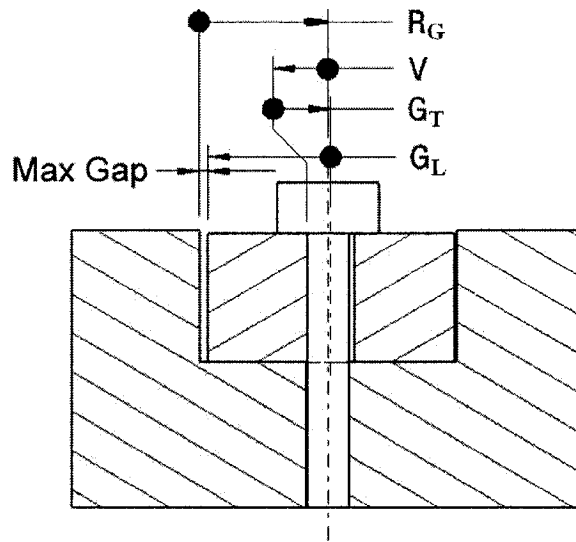


Figure 3.16. Maximum gap configuration for the analysis

$$\text{Average\_gap} = R_G - V - G_T + G_L \quad (3.6)$$

$$\text{Average\_gap} = 0,753 - 0,720 - 0,1227 + 0,1430 = 0,0533 \quad (3.7)$$

$$\text{Sum\_tol} = 0,002 + 0,002 + 0,0093 + 0,0075 = 0,0208 \quad (3.8)$$

$$\text{Min\_gap} = 0,0533 - 0,0208 = 0,0325 \quad (3.9)$$

$$\text{Max\_gap} = 0,0533 + 0,0208 = 0,0741 \quad (3.10)$$



### 3.7.2 Resolution by the Suggested Approach

The previous section demonstrated how to calculate the minimum and maximum gaps between the rail and the block using a traditional uniaxial analysis. This section will demonstrate how to obtain the same results using the suggested approach.

The first step is to identify the topological elements for every part. The software does this by giving an identification number when the geometry is created. A simplified version of surface identification is illustrated in Fig. 3.17 for the block-on-rail assembly. The part graph is created automatically as annotations are added. The rail, block and screw part graphs are presented in Fig. 3.19, 3.20 and 3.21 respectively. The graph construction follows the procedure described in section 3.6.1.

Fig. 3.18 shows the assembly graph. There is a plane-to-plane constraint between surface g6 of the block which lies on surface r8 of the rail. Each screw has an axis-to-axis constraint with the holes of the rail (axis1 on screw1 with axis13 on the rail and axis1 on screw2 with axis12 on the rail) and a cylinder-to-cylinder constraint with inner surface of the holes on the block (V1 on screw1 with g7 on the block and V1 on screw2 with g8 on the block). The control function is defined between surface r7 on the rail and surface g4 on the block. All of the information required to generate the kinematic loops is available. The three kinematic loops identified by the software are represented in Fig. 3.18.

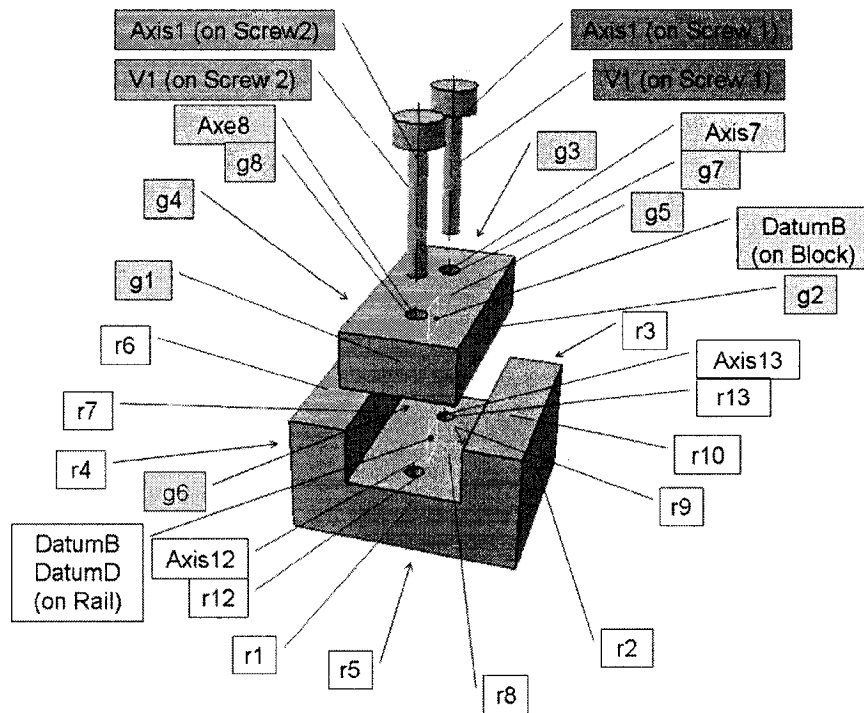


Figure 3.17. Identification of the topology for each part of the block on rail assembly

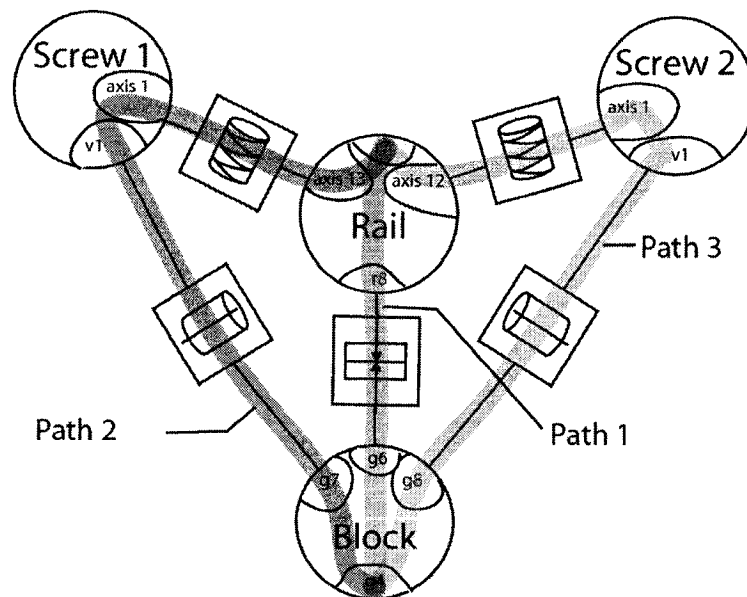


Figure 3.18. Representation of the assembly graph for the block on rail assembly

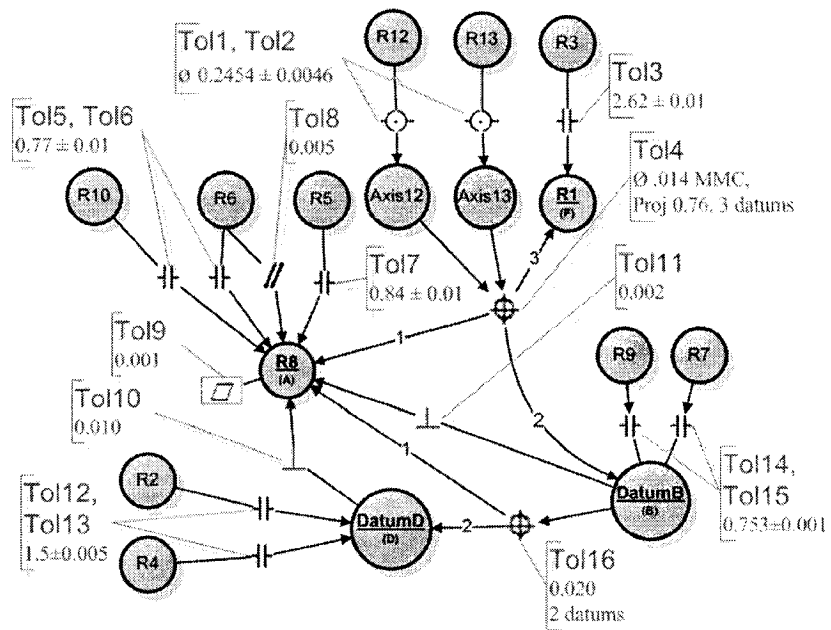


Figure 3.19. Representation of the part graph for the rail

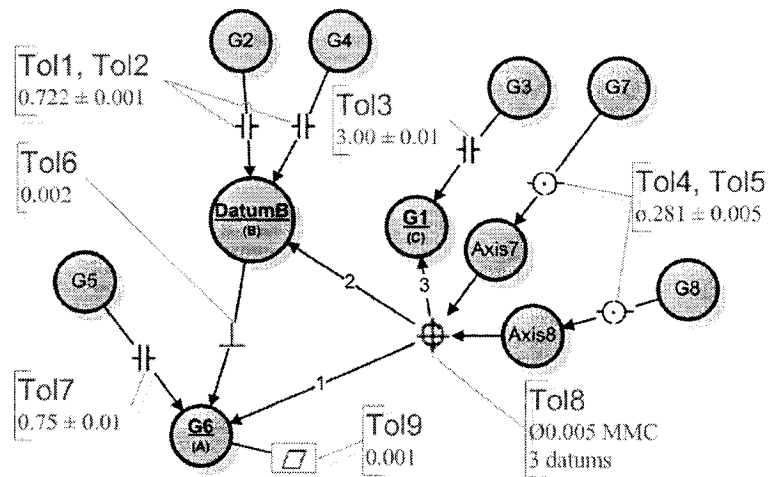


Figure 3.20. Representation of the part graph for the block

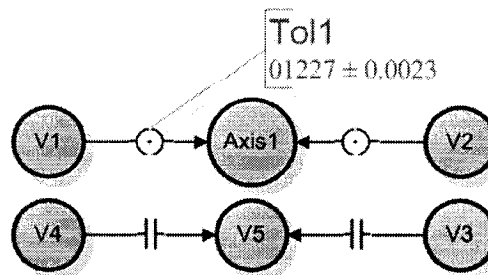


Figure 3.21. Representation of the part graph for a screw

### 3.7.2.1 Tolerance Analysis

Table 3.1 lists the transformations between the topological geometric elements of the kinematic loop. For example, segment  $C_1$  of path 2 indicates a transformation between element R7 and axis 13. It is followed by constraint  $L_1$  which is between axis1 on screw2 and so on. The software generates the loops automatically from the part graphs, the assembly graph, and control function parameters.

Table 3.1. The transformations for the three paths of the block on rail assembly analysis

Path 1	Path 2	Path 3
<ul style="list-style-type: none"> <li>- <math>C_1</math>: R7→R8</li> <li>- <math>L_1</math>: Plane-to-plane (R8→G6)</li> <li>- <math>C_2</math>: G6→G4</li> </ul>	<ul style="list-style-type: none"> <li>- <math>C_1</math>: R7→Axis13</li> <li>- <math>L_1</math>: Axis to Axis constraint (Axe13→Axis1 on Screw2)</li> <li>- <math>C_2</math>: Axis1→V1</li> <li>- <math>L_2</math>: Cylinder-to-Cylinder (V1→G7)</li> <li>- <math>C_3</math>: G7→G4</li> </ul>	<ul style="list-style-type: none"> <li>- <math>C_1</math>: R7→Axis12</li> <li>- <math>L_1</math>: Axis to Axis (Axis12→Axis1 on Screw2)</li> <li>- <math>C_2</math>: Axis1→V1</li> <li>- <math>L_2</math>: Cylinder-to-Cylinder (V1→G8)</li> <li>- <math>C_3</math>: G8→G4</li> </ul>

Only two of the three loops are relevant because the analysis method does not consider the effect of angular error. The first loop has no effect in a uniaxial analysis because the plane-to-plane constraint is normal to the direction of the control function. Moreover, the degrees of freedom remaining on path 1 do not limit the movement of block in the direction of the control function. It is impossible to calculate the minimum or maximum spacing between R7 and G4 using path 1 alone. It would be necessary to use a 3D tolerance analysis method, which would also take into consideration the effect of the mechanism's parallel architecture. Furthermore, loops 2 and 3 will obtain the same result in a uniaxial analysis because the angular error is not considered. Therefore, only the analysis for loop 2 will be elaborated.

### 3.7.2.2 Analysis of the second loop

Fig. 3.22 shows the path of loop 2. The difference between this path and the one obtained by the traditional approach lies with the assembly constraint. The automatic generation of the kinematic loops for the tolerance analysis methodology is possible

because assembly constraints describe the mating relationship between the parts. Loop 2 is a set of transformations, labeled with a “C” if it is within a part and with an “L” if it is an assembly joint. Table 3.2 identifies the transformation, the geometrical elements that are linked by it, and in this case since it is a uniaxial analysis, the dimension of the transformation rather than transformation matrices. The result for each transformation is explained in the next section.

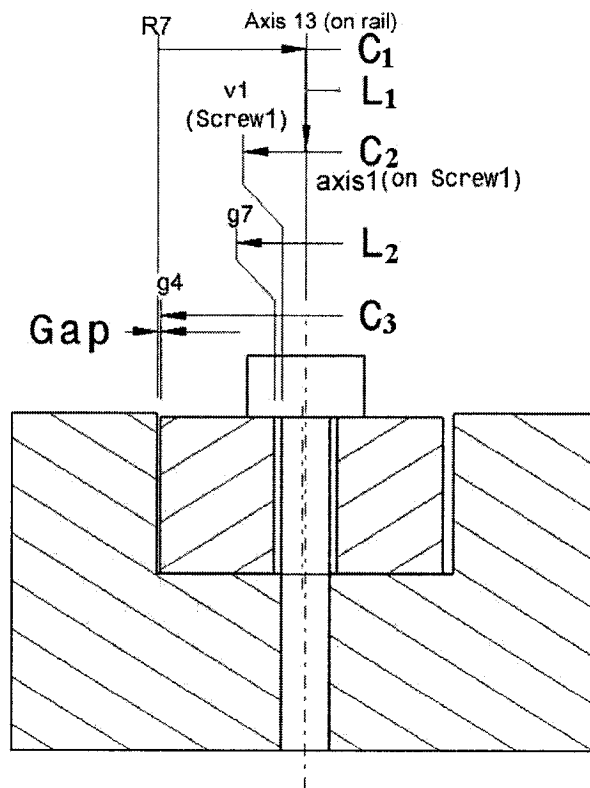


Figure 3.22. The transformations of path 2

Table 3.2. The transformations of Path 2

Transformation	Transformation is between :	Dimension
C <sub>1</sub>	R7→Axis13	0,7530 ± 0,0090
L <sub>1</sub>	Axis13→Axis1 (Screw1)	0,0000
C <sub>2</sub>	Axis1→V1	0,1227 ± 0,0023
L <sub>2</sub>	V1 → G7	G <sub>min</sub> = 0,0356 G <sub>max</sub> = 0,0000
C <sub>3</sub>	G7 → G4	G <sub>min</sub> = 0,5820 ± 0,0095 G <sub>max</sub> = 0,5770 ± 0,0095

In the uniaxial analysis, the assembly constraint  $L_2$  and transformation  $C_3$  have two extreme values:  $G_{\max}$  which will result in the maximum clearance and  $G_{\min}$  which will result in allowance.

Sum of the tolerances in the loop ( $\sum \text{tol}$ ) is:

$$\sum \text{tol} = 0,0090 + 0,0023 + 0,0095 = 0,0208 \quad (3.11)$$

Therefore the gap is:

$$\text{Gap} = C_1 + L_1 - C_2 - L_2 - C_3 \pm \sum \text{tol} \quad (3.12)$$

$$\text{Min\_gap} = 0,7530 + 0,0000 - 0,1227 - 0,0356 - 0,5820 - 0,0208 \quad (3.13)$$

$$\text{Min\_gap} = -0,0081 \quad (3.14)$$

$$\text{Max\_gap} = 0,7530 + 0,0000 - 0,1227 - 0,0000 - 0,5770 + 0,0208 \quad (3.15)$$

$$\text{Max\_gap} = 0,0741 \quad (3.16)$$

### 3.7.2.3 Explanation of the Results

Transformation  $C_1$  defines the relative positioning of Axis13 to surface R7 on the rail.

The rail's part graph makes it possible to know the relation between R7 and axis 13, including all the dependencies. Fig. 3.23 illustrates the dependences for R7 and axis13.

Surface R7 has a dimensional tolerance from Datum B which in turn must respect a perpendicularity tolerance relative to R8, and a position tolerance from R8 and Datum D. Axis13 is positioned relative to Datum B, R8 and R1. The graph shows that Axis12 and R7 are positioned relative to Datum B. This means that subsequent positioning of Datum B does not affect this analysis, but that it is still important to control the orientation. The orientation is controlled by the perpendicularity tolerance of Datum B towards R8 which defines a more strict orientation than the positioning tolerance alone.

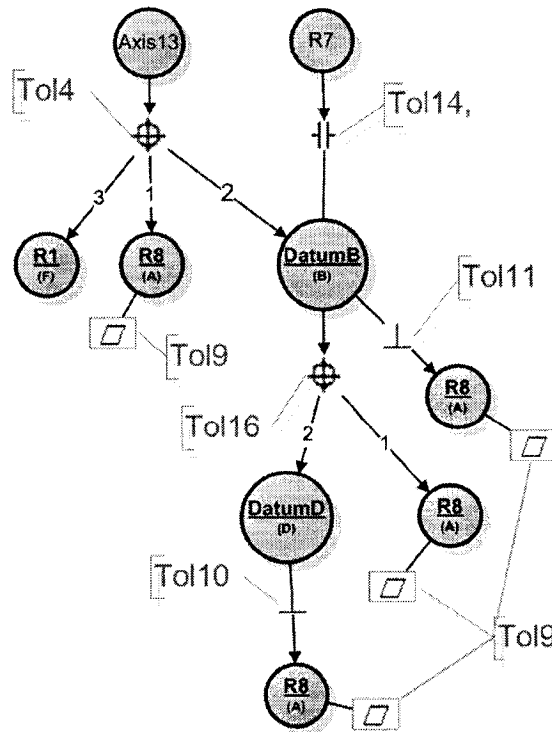


Figure 3.23. Determining the relative positioning between Axis13 and R7.

The distance between R7 and Axis13 is calculated by following the loop in Fig. 3.24. The nominal distance is 0,753-0,0. The tolerance is the sum of the existing tolerances on R7, Axis13 and Datum B ( $0,001+0,001+0,007$ ). The arrows point in the direction of the dependency. For example, Datum B is dependent of Datum D because it is positioned relative to Datum D. The boxed items are those that are not affecting the end result. Datum D is boxed because it is not part of the loop. Also, the position tolerance of Datum B with respect to Datum D (also boxed) has no effect on the orientation control because there is a stricter perpendicularity tolerance on Datum B.

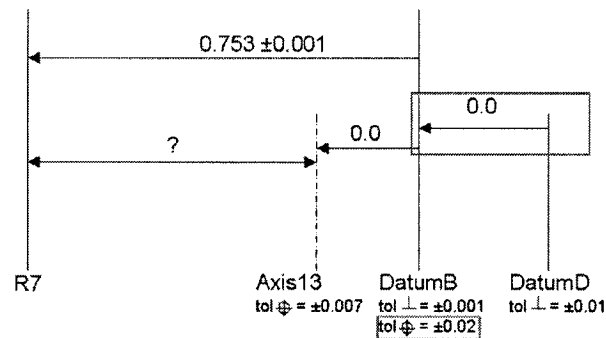


Figure 3.24. Calculating the relative positioning between R7 and Axis13

Assembly constraints do not have tolerances. The tolerances are already considered on the part. The axis-to-axis constraint is a perfect connection used to represent the screw and its hole. Transformation  $L_1$  implies that there can be no displacement between Axis13 and Axis1 of screw 1.

Transformation  $C_2$  is trivial since it corresponds to the radius of the screw ( $0,1227 \pm 0,0023$ ).

Transformation  $L_2$  is a cylinder-to-cylinder constraint between the cylindrical surface of the screw (V1), whose nominal diameter is 0,2454 mm, and the inner surface of the hole on block (G7), whose nominal diameter is 0,281 mm. This means that the maximum distance between V1 and G7 can reach up to 0,0356mm.  $L_2$  is an assembly constraint and so there are no tolerances on it. The effect of the dimensional tolerance on the screw's radius is already considered in transformation  $C_2$ , and the position tolerance on the axis of the screw (which coincides with that of the hole of the rail) is considered in transformation  $C_1$ . Likewise, the dimensional tolerance on the block's hole, the positioning of its axis and the effect of the MMC condition are considered in transformation  $C_3$ .



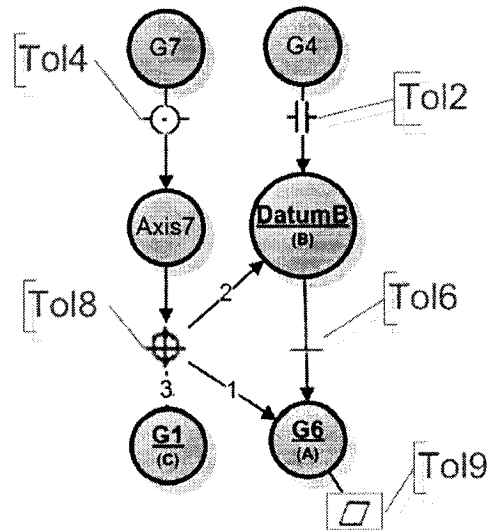


Figure 3.25. Determining the relative positioning between G7 and G4

For transformation  $C_3$ , it is necessary to know the relative positioning between G7 and G4. G4 is located  $0,720 \pm 0,001$  from Datum B. Datum B has a perpendicularity tolerance to G6 of 0,002. G7 is a cylindrical surface of radius  $0,1405 \pm 0,0025$  from Axis7. Axis7 is positioned in MMC from Datum B, G6 and G1 with a tolerance of  $\pm 0,005$  (Fig. 3.25). The material modifier gives a characteristic to the hole by allocating a position tolerance bonus depending on the actual radius of the hole. This maintains the virtual state of the hole at  $0,1430 \pm 0,0025$ . In this case, at nominal, the position tolerance bonus is of  $\pm 0,0025$ . It is necessary to determine the configurations which generate the minimum and maximum gap on the control function. The configurations are illustrated in Fig. 3.26 and 3.27.

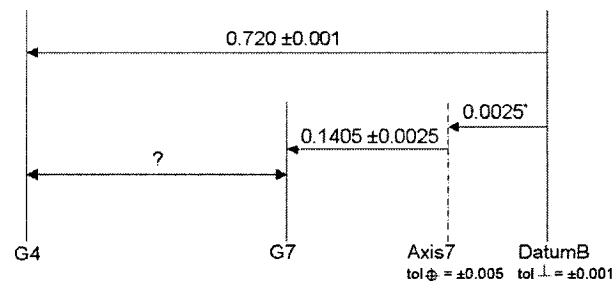


Figure 3.26. Minimum gap calculation of the relative positioning between G7 and G4 (used in the maximum gap calculation of the control function)

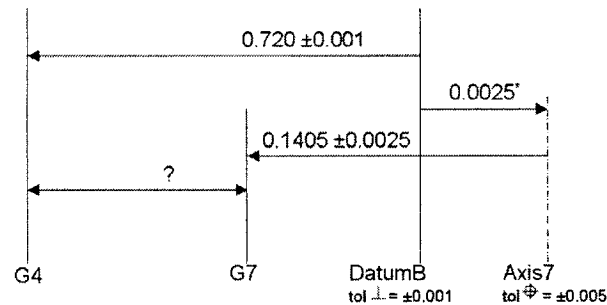


Figure 3.27. Maximum gap calculation of the relative positioning between G7 and G4 (used in the minimum gap calculation of the control function)

### 3.8 Comparison of the Two Methods

The same results were obtained with the traditional uniaxial method and the proposed method of interpretation combined with a uniaxial analysis. However, there are several differences between the two methods.

The first difference is that with the traditional method, a good comprehension of the mechanism is necessary in order to determine the way in which the parts assemble to produce a worst case scenario. This means that the tolerance engineer must figure out how the mechanism works. This analysis is avoided in the proposed interpretation method by the use of assembly constraints. For the purpose of this article, the application of the assembly constraint was considered to be part of the tolerance analysis methodology, but in fact they would simply be imported from a previous phase of the design cycle. The difference in the kinematic loops between the two methods is due to the fact that assembly constraints are being used rather than considering various assembly configurations. Using assembly constraints makes it possible to automate the generation of kinematic loops.

The second difference is at the analysis level. In the traditional approach it was necessary to analyze the detail drawing as a whole, and to pose a certain number of assumptions. The part graphs in the tolerance interpretation method indicate the dependences of each segment and the analysis process is facilitated. However, it is still necessary to pose

some assumptions at analysis level depending on the resolution method. It is obvious that the information provided by the tolerance interpretation method is not used to its full potential when combined with a uniaxial analysis resolution.

Three kinematic loops are possible for this analysis. A 3D analysis should consider that each loop constrains the DOF of the other loops. The final result should be the intersection of the results obtained individually by each kinematic loop. Amongst other things, what is not seen in a uniaxial analysis is the rotational movement of the block around the axes of its holes.

A second phenomenon which has been discussed but was not taken into consideration is the effect of constraint precedence. In the example chosen, the only geometric element where there could have been constraint precedence is on Datum B of the rail, because of the position and perpendicularity tolerances. In this case, the position tolerance was ruled out because it was not part of the loop in the dependency graph. It is important to note that the tolerance interpretation model makes it possible to model this phenomenon.

Another phenomenon which is not managed by a uniaxial analysis, but for which information is available through the interpretation method, is the order of application of the datums and the DOF that are blocked by each one. For example, the position tolerance applied to the axes of the holes on the block has three datum references, A, B and C. In a uniaxial analysis, only movement in the analysis direction is considered, but for a 3D analysis, the datum order of application has an effect because angular errors need to be considered and each datum is constraining different DOF.

The worst angular cases for functional surfaces and datum elements have to be defined in order to consider angular errors during the analysis. In a uniaxial analysis, only the translational worst cases are considered and it generally consists in using the MMC or the

LMC dimensions. For a 3D worst case analysis, it is necessary to consider the amplitudes about each DOF.

### 3.9 Prototype Application for 3D Tolerance Analysis in a VR Environment

The suggested method for tolerance interpretation provides a mechanism to store tolerance information, to generate the kinematic loops and determine the dependencies for any topological element. In order to achieve the maximum potential from this approach, it is necessary to couple it with a 3D tolerance analysis method. Work has already been started on this task.

Fig. 3.28 illustrates a prototype of the 3D tolerance analysis module. The assembly is made up of three triangular prisms inserted inside a “C” shaped support. In the example of Fig. 3.28, the height and width of each prism are equal and the tolerance on the height corresponds to 12.5% of the height. Therefore the stack-up angular error is very large. The umbrella shape corresponds to the volume swept by the upper tolerance zone of the third prism. It represents, in one volume, all the possible cases that could occur given the worst case assembly conditions considering the six DOF's.

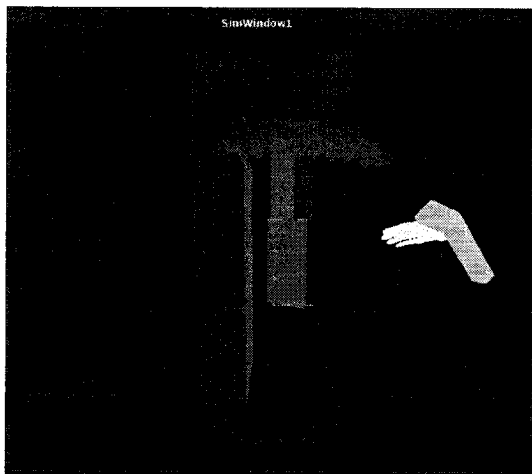


Figure 3.28. Prototype application of a 3D graphical tolerance analysis module in VR

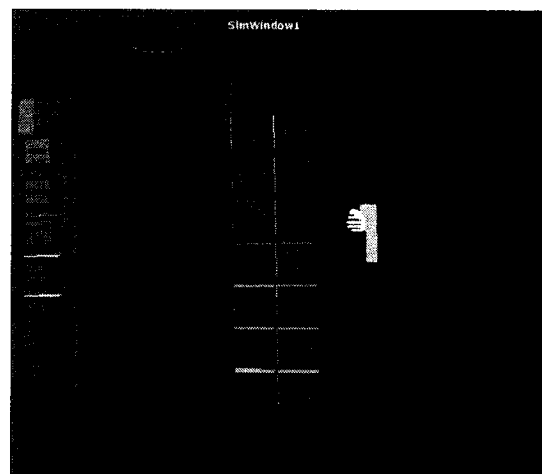


Figure 3.29. Part of the user interface of the prototype

Fig. 3.29 is a preview of the platform's interface. The software is under development to execute in a CAVE virtual reality environment. These images were generated by executing the application in simulation mode. Future work will include a methodology to define the geometry referentials and variation matrices, as well as the implementation of a 3D analysis tool based on the generated tolerance volumes.

### **3.10 Conclusion**

In conclusion, we have established a new methodology to perform tolerance interpretation. A software application was developed to support this methodology and it will eventually support a larger framework that incorporates a 3D analysis solver.

Tolerance interpretation begins with the construction of part graphs which indicate the relations between the geometric elements of a part. With this graph it is possible to determine the dependencies of any geometric element. A second graph, the assembly graph, represents the relations between the parts and makes it possible to determine the kinematic loops. In order to carry out an analysis of the tolerances, it is necessary to consider the kinematic loops and the dependencies for each topological element of the loops.

The examples showed how to apply the two concepts. The block-on-rail case study demonstrates the automation of the interpretation of tolerances and highlights the simplifications of a uniaxial analysis. The next step is to develop the 3D analysis tool.

**CHAPITRE 4: ARTICLE II: A C++ LIBRARY FOR THE  
AUTOMATIC INTERPRETATION OF GEOMETRICAL AND  
DIMENSIONAL TOLERANCES**

Soumis à *International Journal of Advanced Manufacturing Technology*  
Édition *Springer*

Remy Dionne, Luc Baron PhD, Christian Mascle PhD

Department of Mechanical Engineering  
Ecole Polytechnique de Montreal  
C.P. 6079, succ. CV, Montreal, Quebec, Canada

Communicating author:

Remy Dionne

Email : [remy.dionne@polymtl.ca](mailto:remy.dionne@polymtl.ca)

Telephone: (514) 940-2949 ext. 5158

Fax: (514) 940-1399

#### 4.1 Abstract

The first step of a geometrical and dimensional tolerance analysis is to determine the kinematic loops that affect the control function, and the tolerances that influence the relative positioning between two topological elements of a part. This process is called the interpretation of geometric and dimensional tolerances. It is followed by a solution procedure that optimizes the control function. This paper presents the implementation of a C++ library for the interpretation of geometric and dimensional tolerances. The paper first presents the data structures that allow storing the tolerances, the toleranced part data model and the assembly graph model. This is followed by the description of the algorithms used to retrieve the relevant kinematic loops from the assembly graph and the list of tolerances that affect the relative positioning between any two topological elements. A few examples are given to illustrate each case.

#### 4.2 Key words

*Geometrical and dimensional tolerances; analysis; interpretation; ASME Y14.5M; kinematic loops; assembly*

#### 4.3 Introduction

CAD software has contributed considerably in reducing the development time and cost of products over the past thirty years. From a master model of the product, it is now possible to perform many different kind of analysis such as stress, kinematics, ergonomic, simulation of assembly and so on. However, in all these analysis, parts are assumed at nominal dimensions and of perfect geometry. In reality, manufacturing imprecision introduce uncertainties in the dimension and geometry of each individual part. Therefore, tolerances must be applied in order to specify the domain of validity of the parts.

Obviously, there is a high cost associated to allowing tight tolerances, which in turn require high precision manufacturing processes. Conversely, if the tolerances are too

slack, it may not be possible to satisfy the control function, and it may not be possible to assemble the product with those imprecise parts.

The difficulty is that Geometric Dimensioning and Tolerancing (GD&T) requires a methodic approach and the mathematics can be complicated. For example, a 1D tolerance analysis may be relatively simple to undertake: by studying the assembly, the dimensioning and the tolerancing of each part, the analyst is able to determine the kinematic loops that will close the control function and which tolerances must be considered at each point of the loop. This is the interpretation of tolerances and is the most difficult part of the uniaxial analysis. The control function is solved by taking the sum of the 1D vectors that are part of the loop and the sum of the tolerances. This is often accomplished with a tolerancing chart. For a 2D tolerance analysis, the same interpretation work must be done but the mathematics are more complicated and can be solved by hand only for a few trivial cases. Likewise, the resolution of the 3D tolerance analysis is more complicated than the 2D tolerance analysis.

Many people ignore or neglect the importance of performing tolerance analysis and optimization. Part of the reason for this is that a 1D or 2D tolerance analysis will not get results that are precise enough and the 3D methods are still incomplete. Although some 3D tolerancing software are available as commercial products, their modeling schemes do not yet accurately capture all the concepts of GD&T such as the concepts of datum precedence, material modifier conditions and form tolerances among others.

In general, GD&T analysis can be divided into two phases: the interpretation and the resolution. In this article, we will present the implementation of our tolerance interpretation library whose purpose is to capture all the significance of GD&T and make it easily interpretable. The interpretation library can be adapted to work with different resolution methods (1D, 2D or 3D).



#### **4.4 State of the Art**

Traditionally, with uniaxial analysis approaches, the analyst had to study the assembly and the parts, understand how they worked in order to identify the kinematic chains implied in the control function, interpret the significance of the dimensions and tolerances and then solve the problem. This process is prone to error and it is time consuming. Moreover, although the final resolution to the uniaxial problem has the advantage of being simple to solve, it implies many simplifications. Several tolerance analysis tools were developed based upon different approaches.

Schlatter [8] describes how the VSA-3D software was used for the design of an IBM disk drive. In this approach, after the toleranced model of the assembly is created, the analyst must define measurements between the surfaces to be monitored (the control function). Running the tolerance analysis calculation generates a population of possible configurations of the assemblies, each with slightly different constraint values. For each configuration, the measurements are recorded and statistical results are obtained at the end of the analysis. Generating a large number of configurations is required and takes some time to compute. This approach is sometimes called parametric because it studies the effect of the dimensional parameter variations on the measurements.

3DCS is another commercial software application based on the variational concept. An abstract model is built from the master CAD geometry. In the abstract model, the features of the master model are represented by basic geometries such as point, lines, planes, circles, etc. These basic features are defined by parameters. For example, a point is defined by a position vector, a line is represented by two points, and a circle is represented by a point, a radius and a plane. The GD&T annotations define constraints between the basic features of the abstract model. Statistical GD&T analysis is performed by varying the values for the parameters of the basic features and the constraints between them.

Others have developed vector loop assembly model based approaches. It generally involves constructing an assembly graph from which vector loops are extracted and analyzed. A vector loop is a set of vectors that will close a control function. Merkley and Chase [27, 28] have several publications on the subject and resolve for the control function with a matrix-based direct linearization method. The work of Dr. Chase developed through the ADCATS program is the basis for the initial version of Sigmetrix's CETol Six Sigma software for tolerance analysis. Desrochers, Laperrière and Ghie [10,12,13] use a variation twist and jacobian approach to resolve the problem. Their research is based upon the work of Desrochers who introduced the concept of Technologically and Topologically Related Surfaces (TTRS) [11] for the purpose of tolerance synthesis. These methods apply to the resolution of the control function.

Wang [35] proposes to generate the vector loops by testing each dimension individually and verifying their individual effect on the control function. If a change in the dimension also changes the control function, then this dimension is part of a vector loop. Once the dimensions that influence the control function are found, it allows him to find the ones that form loops. One drawback of this approach is that the calculation time increases in function with the number of control functions and the number of dimensions in the assembly. This method is designed to work with toleranced vectors. It is not specified how geometric tolerances are handled with this method.

Although these researches have solid mathematical foundation for solving the tolerance analysis problem, they do not provide a complete solution on how to interpret the tolerance language or how to store the tolerance information in data structures. Particularly, the ASME Y14.5M-1994 standards [29,30] describe that toleranced surfaces can be dependent upon datum surfaces. Datum surfaces can also be toleranced so they also contribute to the error depending on which Degrees of Freedom (DOF) they are blocking. Precedence of constraints, material modifiers and form tolerances are some of the long lasting issues that still have to be resolved by commercial software.

The means by which the tolerance information is stored into the data structure plays an important role in the automatic interpretation and resolution. Moroni and Requicha [17] developed a data structure which represents features, surfaces, datums and tolerances inside a hierarchical graph called the VGraph. This concept differs from that of TTRS since the graph is adapted to represent existing dimensioning whereas the concept of the TTRS is used to define dimensioning. Moreover, the VGraph approach considers the part individually of its function in the assembly whereas the TTRS approach requires the assembly. A similar structure is also proposed by Shah [19] in which a dimension and tolerance graph is created for each control direction (x, y and z). In this graph, the nodes represent the geometric entities and the arcs represent geometric tolerances and dimensions. It was later refined by Kandikjan, Shah and Davidson [20] who combined the three graphs into one and defined clusters to group the constrained entities. This structure was used to perform validation checks on the GD&T and verify the consistency with the dimensioning and tolerancing practices.

Lin and Zhang [3] demonstrated that tolerances can be further optimized if the manufacturing errors due to machine and the process are also considered in the analysis.

#### **4.5 Implementation of the C++ Library**

The library holds the basic function required to build and manipulate a toleranced model. The tolerance model is built separately from the CAD system and defines an application layer on top of topological data, such that the tolerance model created by the library references the native CAD model. The library is used in the following manner: a global method provides an entry point by creating an empty toleranced assembly model. Toleranced parts are added and assembly constraints are defined. The part container stores the tolerance relationships between the topological elements. The assembly container stores the relationship between the parts. The user defines one or more control functions that he wishes to analyze. The kinematic loops, or paths, that close the control

function, are extracted from the assembly graph. Each path is described by a sequence of topological elements and assembly constraints that connect the two elements of the control function. Finally, the tolerance dependencies for each topological element of the path are extracted from the part objects. So, the result of the tolerance interpretation is all the paths that must be considered and the tolerances that apply on each element of these paths. This section describes the containers and the methods that make up the library.

There are seven important data structures in our GD&T library: the *Tolerance* class, the *Modifier* class, the *Datum\_System* class, the *Topology* class, the *Part* class, the *Constraint* class (to define assembly joints) and the *Assembly* class. The UML diagram in Fig. 4.1 illustrates the relationship between the classes.

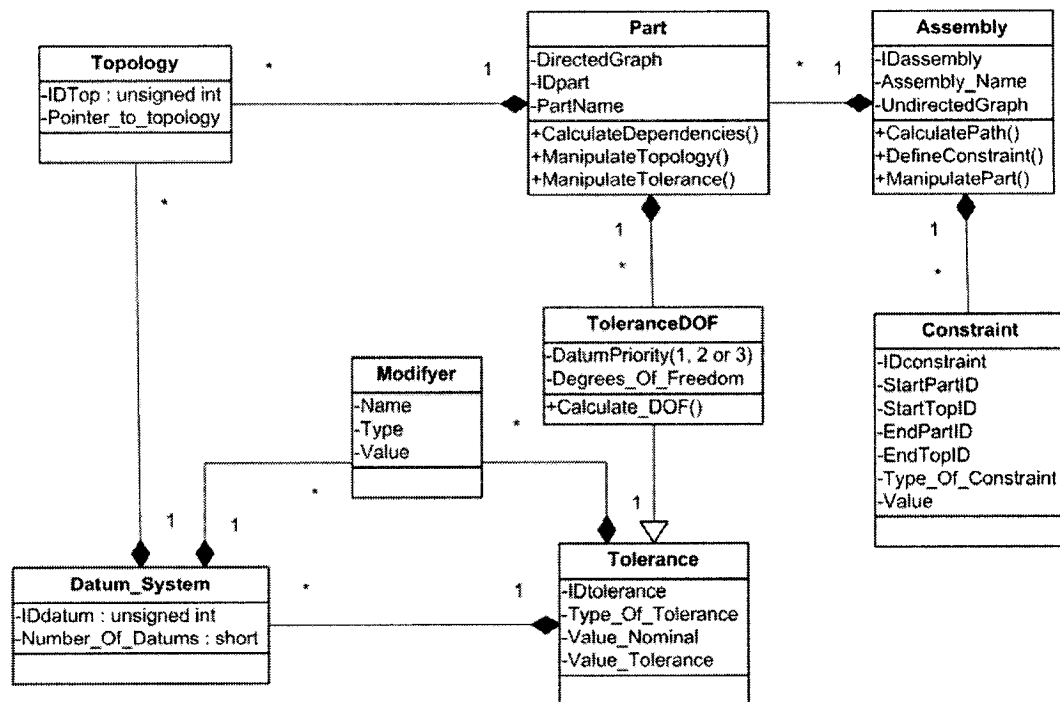


Figure 4.1. UML diagram representing the main classes of the tolerance library

#### 4.5.1 Tolerances

Figure 4.2 illustrates the components of a geometrical tolerance. The arrow points to the element on which the tolerance applies. In the first box from the left, the symbol

represents the type of the tolerance. The second box indicates the tolerance value (diameter of 0,005) which may be accompanied by one or more modifiers (in this case, there is only one and it is for MMC). Finally, the remaining boxes define the datum system. In this example, datums “A”, “B at MMC” and “C” make up the datum system. Not all tolerances have a datum system and if they do, the number of datums can vary from one to three. When there is more than one datum, the order of the datum affects how the degrees of freedom are constrained.

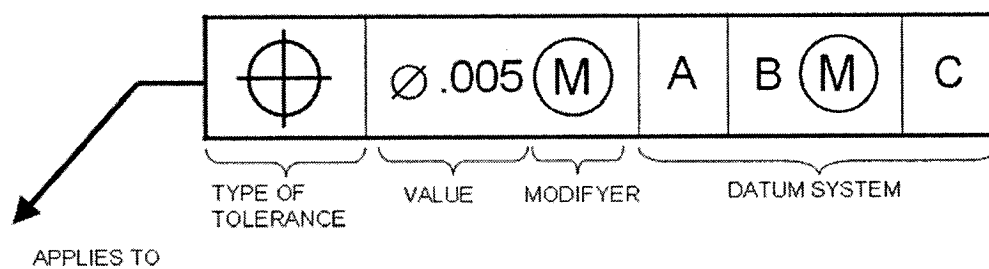


Figure 4.2. Components of a geometrical tolerance

Several classes were defined in order to model the tolerances: *Tolerance*, *Modifier*, *DatumSystem* and *ToleranceDOF*.

The *Tolerance* class has data members to specify the type of the tolerance, the nominal value (for dimensional tolerance), the tolerance value, a list of modifiers and a datum system. The *Modifier* class has two attributes which are the type of modifier and a value.

The supported modifiers conform to the ASME Y14.5M-1994 standard which are: MMC, LMC, projection, free state, tangent plane, diameter, sphere diameter, radius, controlled radius, reference, arc length, statistical tolerance, between and none (if no modifier is specified).

The *Datum\_System* class is made up of an identification number, an attribute to specify the number of datums in the datum system, a list of topology object ids, and a list of

modifier. In the example above, there are three datums. Datums A, B and C refer to topological elements. A and C do not have any modifiers but B has a MMC modifier. So, the list of modifiers is of size 3 and datums A and C have the “none” modifier type while datum B has the “MMC” modifier type.

The relationships between the classes are more clearly indicated on Fig. 4.1. The *Datum\_System* class and the *Tolerance* class aggregate a *Modifier* object. The *Tolerance* class also aggregates a *Datum\_System* object. The *Datum\_System* object aggregates the *Topology* objects that compose it.

#### 4.5.2 Part Class

As shown in Fig. 4.1, the *Part* class has three data members which are an identification number, a name and a directed graph. The *Topology* and the *ToleranceDOF* objects are aggregated to the *Part* class as being the nodes and edges of the directed graph. The purpose of the *Part* class is to store the tolerance data of a part and provide a mechanism to determine the tolerance dependencies of any topological element.

The directed graph is instantiated using the Boost Graph Library [36]. This library provides the means to manipulate the graph data in a manner similar to what the Standard Template Library (STL) does for C++ lists and vectors. Methods allow scanning through the list of nodes, the list of edges, the neighbors of a node, the out edges or in edges of a node and so on. Furthermore, it is a template library so the nodes and edges of the graph can be of any user defined type. In this directed graph, the nodes are *Topology* objects and the edges are *ToleranceDOF* objects.

The *Topology* object is required to provide a reference to the actual CAD data model. Its two data members are an identification number and a pointer to CAD topology.

The *Part* class provides methods to define and manipulate the nodes (topology) and the edges (tolerances) of the graph. A topology node is created for each tolerated topological element of the part. The tolerance information is stored in the edges of the part object's directed graph. The topological element on which the tolerance applies is the starting node of the edge. An edge is created between the starting node and each datum. In case the tolerance is independent, the edge starts and ends on the same node. *ToleranceDOF*, the object attached to the part graph's edge, inherits from the *Tolerance* class, specifies the datum priority (datum A is priority 1, B is priority 2 and C is priority 3), and the DOF allowed by the tolerance. Dimensional tolerances are also stored in the *ToleranceDOF* object, but the edge is created as bi-directional and the distribution of the tolerances on the surfaces is considered during the resolution stages.

In the *Part* class, there is a method to determine the dependencies of a given topological element. This is done by obtaining the dependency tree of a node of the graph as is explained in section 4.6.2.

### 4.5.3 Assembly Class

The relationship between parts is described by the *Assembly* object. The *Assembly* class has three data members: a name, an identification number and an undirected graph object created using the Boost Graph library. Instances of the *Part* class make up the nodes of the undirected assembly graph. The edges are kinematic joints that model the relative movement between two topological elements on different parts. Three types of assembly constraint are defined in this article (in Fig. 4.3), for the purpose of demonstration, but there are others. The reader may refer to [12] or [13] for other twist-based assembly constraint definitions.



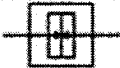





Constraint	Symbol	Free DOF	Constrained DOF	Torsor
Axis to axis		None		$\begin{Bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{Bmatrix}$
Plane to plane				$\begin{Bmatrix} \infty & \alpha \\ \infty & \beta \\ \omega & \infty \end{Bmatrix}$
Cylinder to cylinder				$\begin{Bmatrix} \infty & \infty \\ v & \beta \\ w & \delta \end{Bmatrix}$

Figure 4.3. Kinematics joints defined in the library. A solid arrow indicates a DOF that is free or blocked. A dashed arrow indicates a partially blocked DOF.

I) Axis to axis constraint: The axes of two parts are perfectly aligned. The DOF are blocked and the two parts act as if they were only one part.

II) Cylinder to cylinder constraint: Is used if a peg enters in a hole of equal or larger diameter. The link is made between the two cylindrical surfaces rather than between the axes. The peg is free to move in the axial direction but is limited in its translation and rotation movement about directions that are normal to the hole's axis. The amplitude of movement depends on the diameters and lengths of the hole and the peg.

III) Plane to plane constraint: The normals of two surfaces are aligned and their faces are in contact. However, the parts are free to move in translation in the plane and in rotation around the plane's normal.

The kinematics joints are instances of the *Constraint* class which has the following data members: an identification number, a type of constraint, a torsor with values indicating the allowable movement for each degree of freedom, a starting part id, a starting element id, a target part id and a target element id.



The assembly graph makes it possible to generate the kinematic loops (also called paths) that are related to the control function. A kinematic loop is the sequence of surfaces and assembly constraints that connects the two surfaces of the control function in the assembly graph.

#### 4.6 Interpretation of the Tolerances

The steps of the tolerance analysis process are illustrated in Fig. 4.4. The first step, building the tolerance model, is done as annotations, tolerances and assembly constraints are defined on the parts. This functionality is supported by the data structure of the library. Then the control functions are defined by the user. Next, the interpretation of the tolerances is done in two steps: First, the kinematic loops (also called paths) that define the control function are identified. The paths are obtained from the assembly graph. Then, the tolerance dependencies are obtained for each element of the path using the part graphs. This section presents a use case and explains how the interpretation of the tolerances is achieved. In order to complete the analysis of the tolerances, the interpretation of the tolerances must be followed by a resolution method.

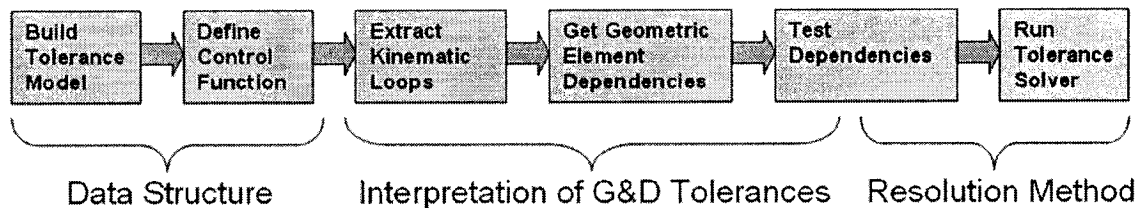


Figure 4.4. The interpretation of the tolerances in the tolerance analysis process

##### 4.6.1 Identification of the Paths

Consider the slider on rail example illustrated in Fig. 4.5. A block with two holes in it, called the slider, is bolted on a rail by means of two screws. Fig. 4.5 also shows the identification for each toleranced topological element. The toleranced model is built from the detail drawings of each part and the assembly constraints that are defined between them. Figure 4.6 shows a detail drawing of the rail in which all the tolerances

have been identified from Tol1 to Tol16. Similar detail drawings for the block and the screws also exist but are not shown. The assembly is modeled as illustrated in Fig. 4.7. The bottom face of the slider (face “G6”) rests on top of the rail (on face “R8”). So a plane to plane assembly joint has been defined between face G6 of the slider and face R8 of the rail. The diameter of the holes in the slider is bigger than the diameter of the shafts of the screws such that there is a gap between them. Therefore, an assembly joint of type cylinder to cylinder was defined between surface G8 of the slider and V1 on the first screw and another one between G7 on the slider and V1 on the second screw. The cylinder to cylinder assembly joint limits the movement between the shaft and the hole. Finally, since the threads on the screw are “grabbed” by the threads on the rail which tends to auto-center the screw, and because tolerances are applied so that this is considered as a fixed fastener assembly, the hypothesis made is that the screw’s axis will coincide with the threaded hole’s axis. The screws are considered to be mounted into the rail. This is why there is an axis to axis constraint between the Axis1 of the first screw and Axis13 on the rail and between Axis1 of the second screw and Axis12 on the rail.

The gap between the sides of the slider and the inner sides of the rail will be verified to ensure the proper functionality of the product. So the user must define a control function between faces R7 on the rail and G4 on the slider. At this point, the data structures and a control function are defined. The next step is the interpretation of the tolerances.

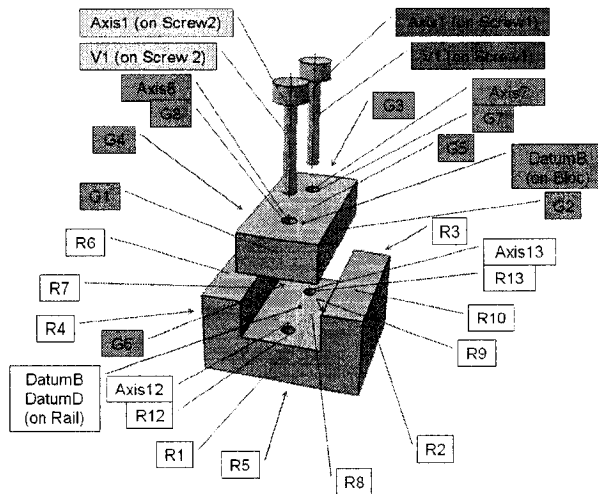


Figure 4.5. Identification of the topological elements of the block bolted on rail assembly

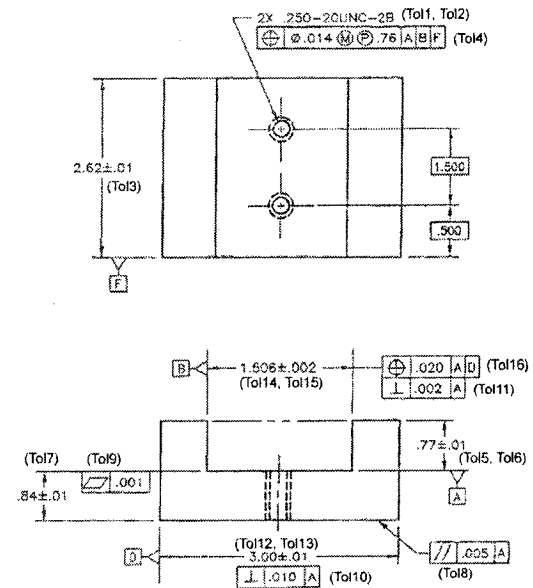


Figure 4.6. Detail drawing of the rail with identification of the tolerances

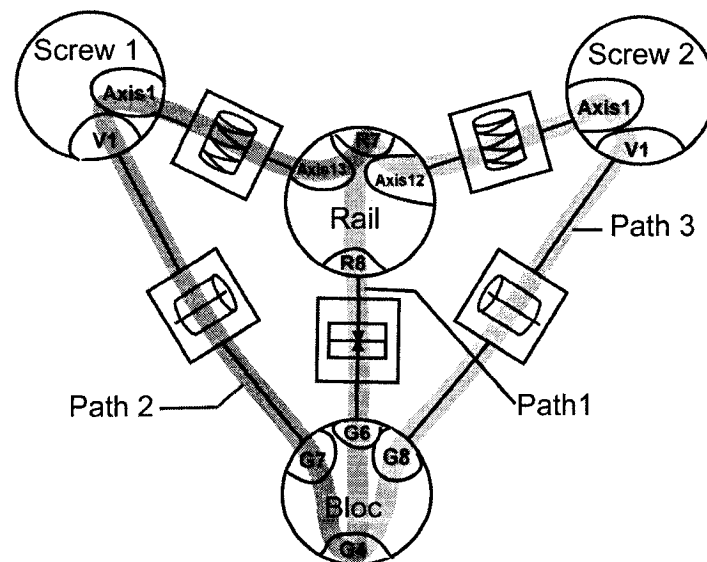


Figure 4.7. Representation of the three paths that close the control function between R7 and G4 on the block bolted on rail assembly

Figure 4.7 also illustrates the three paths that bridge the connectors for R7 and G4. Path 1 closes the control function by passing through the plane-to-plane connection and paths 2 and 3 close the control function by passing through the screws. The transformations involved in each path are summarized in Table 4.1. Path 2 is a transformation from R7 to Axis 13 on the rail, followed by the axis-to-axis joint with the axis of screw 1. Then there is another transformation from the axis of the screw to surface V1 of the screw, followed by the cylinder-to-cylinder joint between V1 and surface G7 on the slider and finally another transformation from surface G7 to G4 on the slider.

**Table 4.1.** The transformations for the three paths of the block on rail assembly

Path 1	Path 2	Path 3
<ul style="list-style-type: none"> <li>- C<sub>1</sub>: R7→R8</li> <li>- L<sub>1</sub> : Plane to plane (R8→G6)</li> <li>- C<sub>2</sub>: G6→G4</li> </ul>	<ul style="list-style-type: none"> <li>- C<sub>1</sub>: R7→Axis13</li> <li>- L<sub>1</sub> : Axis to axis (Axis13→Axis1 on Screw2)</li> <li>- C<sub>2</sub>: Axis1→V1</li> <li>- L<sub>2</sub> : Cylinder to cylinder (V1→G7)</li> <li>- C<sub>3</sub>: G7→G4</li> </ul>	<ul style="list-style-type: none"> <li>- C<sub>1</sub>: R7→Axis12</li> <li>- L<sub>1</sub> : Axis to axis (Axis12→Axis1 on Screw2)</li> <li>- C<sub>2</sub>: Axis1→V1</li> <li>- L<sub>2</sub> : Cylinder to cylinder (V1→G8)</li> <li>- C<sub>3</sub>: G8→G4</li> </ul>

The kinematics loops, are automatically extracted using the «CalculatePaths()» method of the assembly class. This method takes two connectors as input parameters. A connector is a reference to a topological element on a part. The two connectors define the control function. In the slider on rail example, the first connector is surface R7 on the rail and the second connector is surface G4 on the slider.

In order to retrieve the paths, the «CalculatePaths()» method creates a list of all possible paths that start on the first connector and that pass through the second connector on the assembly graph. All the paths that do not contain the second connector (ie: the slider) are filtered out.

The algorithm that accomplishes this task is the following: first, a path that contains only the starting node is created. This path is passed to a recursive method. In this method, a

copy of the current path is created and added to a list of paths for each out edges of the last node of the current path whose target node would create a new path in the list of paths. Then, the target nodes are added to the paths and the new paths are passed to the recursive method in order to examine the out edges of the last node. After all the possible paths have been identified, only the paths that include the target node are kept. If the target node is not the last element of a path, then everything in that path that comes after the target node is truncated.

#### **4.6.2 Retrieving the Tolerance Dependencies of a Topological Element**

The positioning and orientation of topological elements is defined by a transformation matrix. The tolerances that apply on the topological element define the tolerance zone.

The tolerance dependencies for any topological element of a part are determined by building a dependency tree from the part's graph. Let us pursue with the rail part as an example. Figure 4.8 shows the nomenclature used for the part graph representations. Figure 4.9 shows the part graph for the rail that is obtained from the detail drawing. Figures 4.10 and 4.11 show the dependencies for topological elements Axis13 and R7 which are listed in path 2. The dependencies must be determined for all the topological elements of a path (ie: all the elements that are in Table 4.1). They are obtained from the graph by recursively scanning through the out edges of the neighboring nodes. The tolerance dependencies can be used to model the tolerance zone.



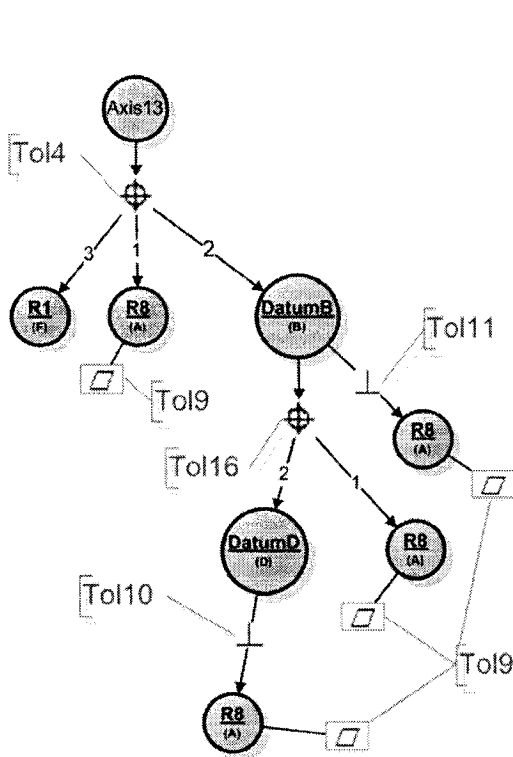


Figure 4.10. Dependencies for Axis13

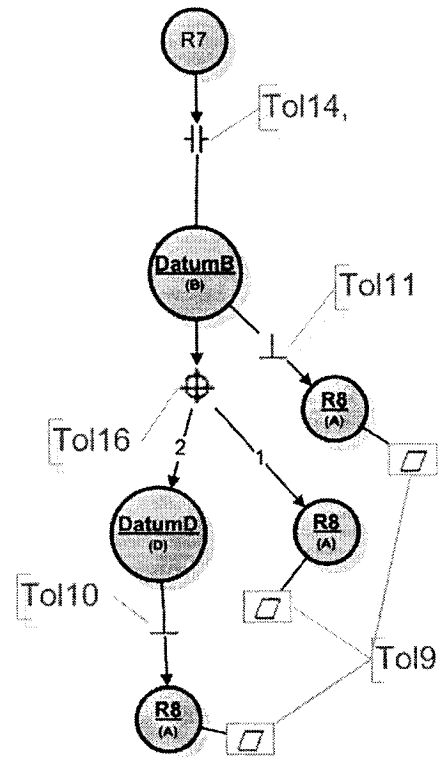


Figure 4.11. Dependencies for R7

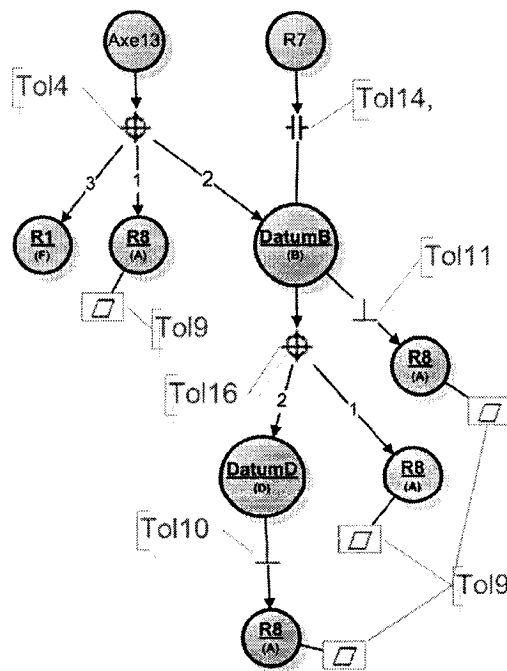


Figure 4.12. Dependencies common to Axis13 and R7

### 4.6.3 Relative Positioning Between Two Toleranced Topological Elements

Each segment of the path is associated to a transformation matrix between two toleranced topological elements. In path 2, there is a transformation from R7 to Axis13 (see Fig. 4.7 and Table 4.1). From Fig. 4.10 and Fig. 4.11, it can be seen that Axis13 and R7 have some common datum references. This is illustrated in Fig. 4.12. Surface R7 has a dimensional tolerance from Datum B, which in turn, must respect a perpendicularity tolerance relative to R8 and a position tolerance from R8 and Datum D. Axis13 is positioned relative to Datum B, R8 and R1. The graph shows that Axis13 and R7 are both positioned relative to Datum B. This means that subsequent positioning of Datum B has no effect for this analysis. However, the orientation is controlled further by the perpendicularity tolerance of Datum B towards R8.

However there are cases in which it is not clear if a tolerance has an effect on the relative positioning between two topological elements. In that case, the sensitivity of the tolerance on the control function should be tested. Wang [35] proposed to calculate the sensitivity of all the tolerances to know if they have an effect on the control function. The problem with that method is that the algorithm is not very efficient in the case that there are many annotations to consider. We propose to test only the annotations common to the spanning tree of the two topological elements. In this case, the sensitivity of the tolerances should be tested on each degree of freedom. Desrochers, Laperrière and Ghie [10,11,13] discussed a way to calculate the effect of tolerances on the control function using a jacobian, a projection matrix and interval mathematics.

### 4.6.4 Method of Resolution

This library provides only the results for the interpretation of GD&T. The next objective is to implement a tolerance solver that would query for results from the library and then calculate the control functions. The results from the interpretation library can also be outputted to a text file. Figure 4.13 shows the output for path 2 of the slider bolted on rail example. Note that parts of the output have been removed to facilitate the understanding.



The output file indicates the three paths and the dependencies for each element of the paths. Figure 4.7, which illustrates the three paths that close the control function, and Fig. 4.10 and Fig. 4.11 which show the dependencies for elements R7 and Axis13, were created from the information of the output file.

```
Source : 0 , target : 1
THIS IS PATH 1
THIS IS A PATH: 0 1
[...]
THIS IS PATH 2
THIS IS A PATH: 0 2 1

R7 on Rail
From : R7 to :DATUM_B
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 9, TolType 15 Value: 0.753, tol value: 0.001, NO
MODIFIER, Value: 0 / Datum 15, NO MODIFIER, Value: 0/

From : DATUM_B to :R8
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO
MODIFIER, Value: 0 / Datum 8, NO MODIFIER, Value: 0/ /
Datum 16, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO
MODIFIER, Value: 0 / Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY
From : DATUM_B to :DATUM_D
DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO
MODIFIER, Value: 0 / Datum 8, NO MODIFIER, Value: 0/ /
Datum 16, NO MODIFIER, Value: 0/

From : DATUM_D to :R8
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 15, TolType 8 Value: 0, tol value: 0.01, NO
MODIFIER, Value: 0 / Datum 8, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO
MODIFIER, Value: 0 / Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY
From : DATUM_B to :R8
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 14, TolType 8 Value: 0, tol value: 0.002, NO
MODIFIER, Value: 0 / Datum 8, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO
MODIFIER, Value: 0 / Datum 0, NO MODIFIER, Value: 0/
-----NO DEPENDENCY
AXE13 on Rail
From : AXE13 to :R8
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 3, TolType 10 Value: 0, tol value: 0.014, MMC, Value:
0 / Datum 8, NO MODIFIER, Value: 0/ / Datum 15, NO
MODIFIER, Value: 0/ / Datum 1, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO
MODIFIER, Value: 0 / Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY
From : AXE13 to :DATUM_B
DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 3, TolType 10 Value: 0, tol value: 0.014, MMC, Value:
0 / Datum 8, NO MODIFIER, Value: 0/ / Datum 15, NO
MODIFIER, Value: 0/ / Datum 1, NO MODIFIER, Value: 0/

From : DATUM_B to :R8
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO
MODIFIER, Value: 0 / Datum 8, NO MODIFIER, Value: 0/ /
Datum 16, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry
DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0}
Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO
MODIFIER, Value: 0 / Datum 0, NO MODIFIER, Value: 0/
[...]
THIS IS PATH 3
THIS IS A PATH: 0 3 1
[...]
```

Figure 4.13. Text file output for the control function R7 on rail to G4 on Slider

The control function can be solved for using a variety of methods. It could be coupled to a 1D, 2D or 3D tolerance analysis solver. So far, the tolerance interpretation results have been used with a uniaxial analysis solver that was adapted to this interpretation of the tolerances. A complete example of the interpretation and uniaxial resolution process is demonstrated in an article called *Methodology for the Interpretation of Geometrical and Dimensional Tolerances in an Assembly* that has been submitted by Dionne, Mascle and

Baron to the *International Journal of Advanced Manufacturing Technology* in August 2007. The same result as in [4] is obtained if the same uniaxial hypotheses are made. The article demonstrated the utility of the approach. It also showed that the uniaxial analysis resolution discards much of the information that is available and that the result obtained is a simplification of the reality.

Future works include developing a tolerance analysis solver in order to automate the entire process and take into consideration the data that is discarded through a uniaxial analysis. We believe Ghie, Laperrière and Desrochers' unified jacobian approach [13] is a viable way to implement such a solver, but further study is still required. Our current research interest is also in developing a 3D graphical method that generates the swept volume where the tolerance zones could exist. A preview of this graphical method is shown in Fig. 4.14 where the umbrella shaped volume on top of the third block represents the space where the tolerance zone of the top face of the part could exist given the tolerance stack-up in the assembly.

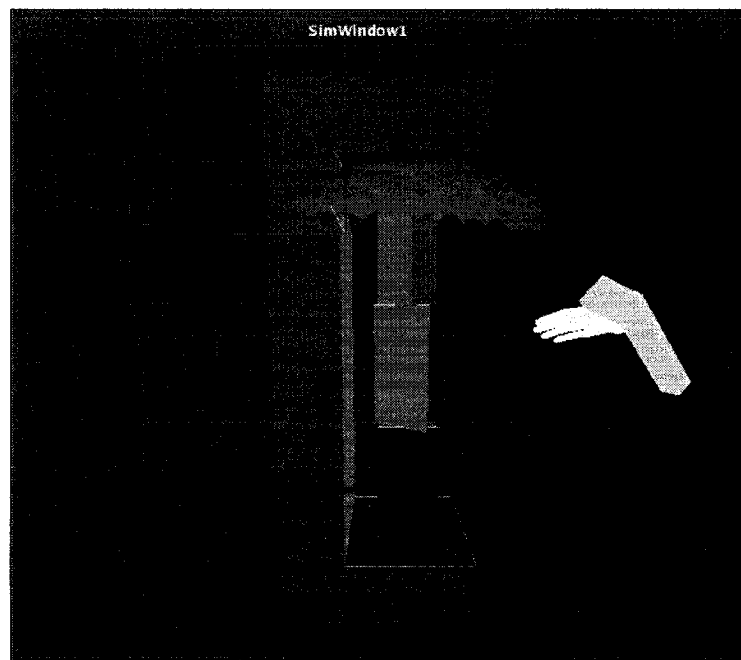


Figure 4.14. Volume obtained using a prototype 3D graphical analysis method

#### **4.7 Conclusion**

This paper presented a C++ library that implements a geometric and dimensional tolerance interpretation method. This library automates the tolerance interpretation process and is to be used in conjunction with a tolerance solver.

The first step of the development process was to establish a data structure to organize tolerance information in such a way that it is easily interpretable. The main features of the data structure are the part graph and the assembly graph. The assembly graph makes it possible to automatically extract the kinematic loops that define the control function. The part graphs are used to determine all the tolerances that may affect the control function. Also, a method is proposed to filter out the tolerances that have no effect on the relative positioning between two elements of the path, although in some cases it is still required to test the sensitivity of the tolerances on the control function.

The tolerance interpretation process can be used to provide a view of tolerance stack-ups on a topological element, as a tool to identify the kinematic loops, or as a filter on the tolerances that should be considered in a tolerance analysis.

## CHAPITRE 5 : DISCUSSION GÉNÉRALE

L'objectif initial de cette recherche était de développer et implémenter une nouvelle approche graphique 3D afin de résoudre la fonction de contrôle. L'étape de résolution s'avère être une des dernière étape dans le processus de l'analyse des tolérances géométriques et dimensionnelles. Plusieurs infrastructures sont nécessaires afin de permettre une automatisation complète de cette méthode de résolution. Entre autre, elle est précédée par une étape d'interprétation des tolérances qui doit générer les boucles cinématiques et aussi capter l'influence de chaque annotation sur la fonction de contrôle. Le but de ce chapitre est, en premier lieu, de présenter les difficultés rencontrées par rapport à la méthode graphique 3D d'analyse des tolérances qui est présentée au chapitre 2. En un deuxième temps, les résultats obtenus lors de l'interprétation des tolérances dont la procédure est décrite au chapitre 3 et l'implémentation au chapitre 4 seront présentés.

### 5.1 Difficultés rencontrées avec la génération des volumes d'existence

Les outils développés permettent de réaliser une analyse graphique du tolérancement directement sur le modèle 3D. De plus, cette analyse prend en considération la grandeur des surfaces tolérancées ce qui permet de considérer la propagation des erreurs d'orientation. La démonstration au chapitre 2 démontre la faisabilité de la génération des volumes d'existence de zone de tolérance. Toutefois, il y a plusieurs limitations à l'application développée. Les difficultés rencontrées ont permis de mieux cerner les infrastructures nécessaires pour supporter une telle application. On note le besoin d'un module pour interpréter la signification du tolérancement. Ce premier besoin justifie en grande partie le sujet de ce mémoire est élaboré à partir du chapitre 3. Ensuite il faut un module pour faire l'analyse des tolérances. L'état de la recherche au chapitre 1, ainsi que certaines sections des chapitres 3 et 4 présentent les recherches effectuées par d'autres chercheurs sur ce sujet. Le chapitre 2 présentait un nouveau moyen d'effectuer cette analyse par génération des volumes d'existences des zones de tolérances.

Une des difficultés dans le développement de l'application en réalité virtuelle était la nécessité d'intégrer et développer plusieurs autres infrastructures requises pour faire l'analyse des tolérances mais qui ne sont pas directement relié au sujet spécifique de cette recherche. Ce sont des fonctionnalités banales dans les modeleurs commerciaux qui devaient être implémentés dans l'environnement de réalité virtuelle. Par exemple, l'intégration d'un solveur de contraintes d'assemblage, et la sélection d'éléments topologique des modèles (face, axes, etc). Dans l'application développée, il était possible de spécifier seulement des contraintes d'assemblage de coïncidence de plan ou coïncidence d'axe, et seulement une seule contrainte pouvait être appliquée par pièce.

La méthode proposée consiste à faire de nombreuses opérations booléennes sur les modèles. La librairie GTS [26] a été utilisé pour cette fonctionnalité. Par contre, de nombreux bogues et limites ont été découvert par rapport à l'utilisation de cette librairie ce qui a limité grandement l'obtention de résultats pour l'approche graphique 3D.

La librairie GTS permet d'effectuer des opérations booléennes sur des modèles composés de facettes triangulaires. Les deux solides doivent être manifold, orientables et il doit y avoir une intersection entre les deux. C'est-à-dire que pour chaque solide, il doit être possible de définir l'espace intérieur et l'espace extérieur. Un solide dont les surfaces frontières s'intersectent n'est pas manifold. Aussi, si une facette du solide est trop petite, des arrondis dus à la précision machine peuvent entraîner le solide à devenir non-manifold suite à une transformation ou une opération booléenne. Certaines arêtes peuvent s'effondrer sur elle même, avoir une grandeur nulle ou exister en double ce qui entraîne le solide à devenir non-manifold. Puisque GTS utilise la normale de chaque facette pour déterminer la direction de la matière, si une ou plusieurs facettes comportent un problème, cela fait échouer le code pour les opérateurs booléens. Les opérateurs booléens dans GTS fonctionnent bien tant que les solides fournis respectent ces trois conditions. Le problème est que GTS ne garantit pas que le solide résultant d'une opération booléenne sera manifold. Les Fig. 5.1 et 5.4 représentent des cas types. Les

Fig. 5.2 et 5.5 montrent les mêmes pièces après que le maillage ait été réparé automatiquement par un algorithme qui élimine les triangles dégénérés.

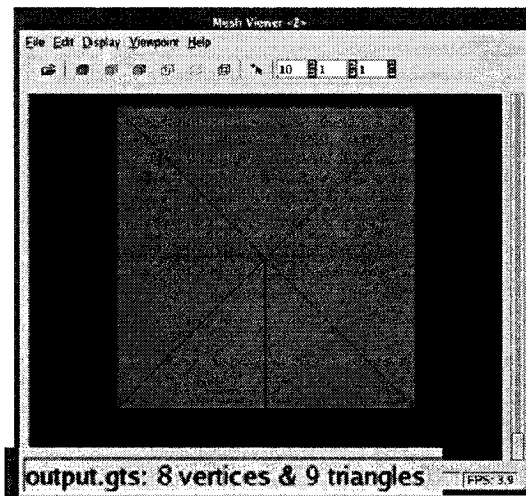


Figure 5.1. Exemple de maillage contenant des triangles dégénérés

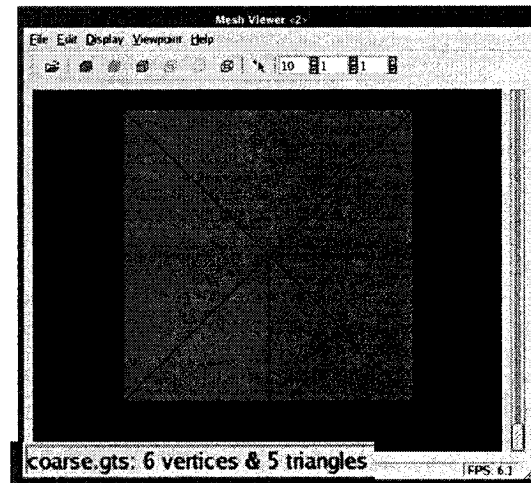


Figure 5.2. Exemple de maillage réparé

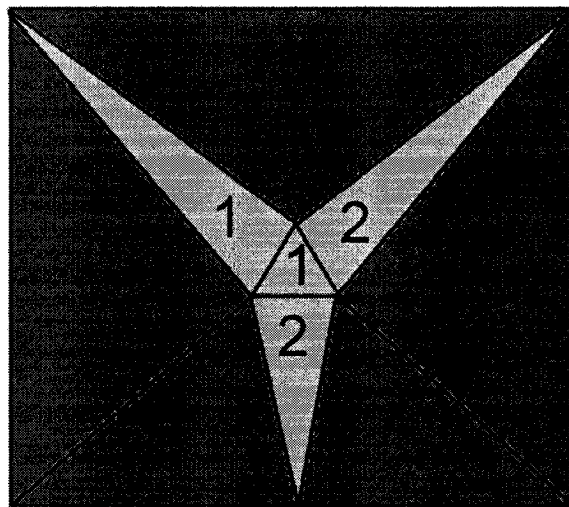


Figure 5.3. Représentation des triangles dégénérés de la figure 5.51

Le nombre de vertex et de triangles est indiqué dans le coin inférieur gauche des figures 5.1 et 5.2. Il y a un très petit triangle au centre de la Fig. 5.1 et trois autres très petits

triangles qui vont du centre vers une des extrémités (en bas, en haut à gauche et en haut à droite). En tout, il y a huit vertex et neuf triangles dans cette image. La Fig. 5.3 exagère les grosseurs des petits triangles. Afin de réparer ces maillages, un algorithme créé, détermine que plusieurs arêtes sont nulles ou très petite et aussi que certains angles sont presque nuls ou très près de  $180^\circ$ . L'algorithme essaie de réparer le maillage automatiquement en détruisant les triangles par paire. Dans ce cas, les triangles identifiés par le chiffre « 1 » à la Fig. 5.3 sont détruits en premier en effaçant l'arête qui est commune aux deux. Les extrémités de cette arête sont combinées en un seul point. Des arêtes deviennent flottantes et sont effacées. Ensuite, la même procédure est appliquée aux triangles qui sont identifiés par le chiffre « 2 ». Le résultat est montré à la Fig. 5.2 et il ne reste que 6 vertex et 5 triangles.

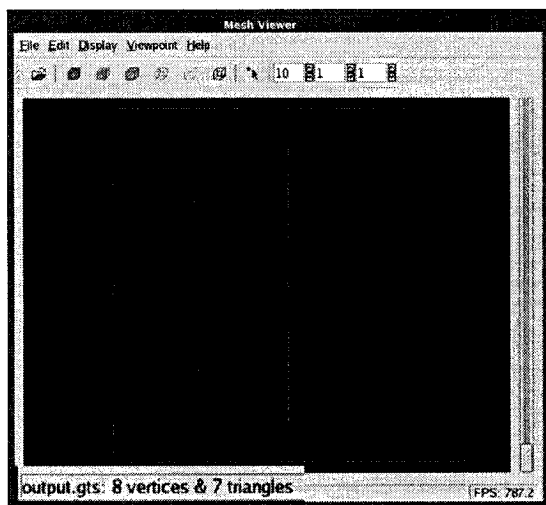


Figure 5.4. Maillage avec triangle dégénéré au centre

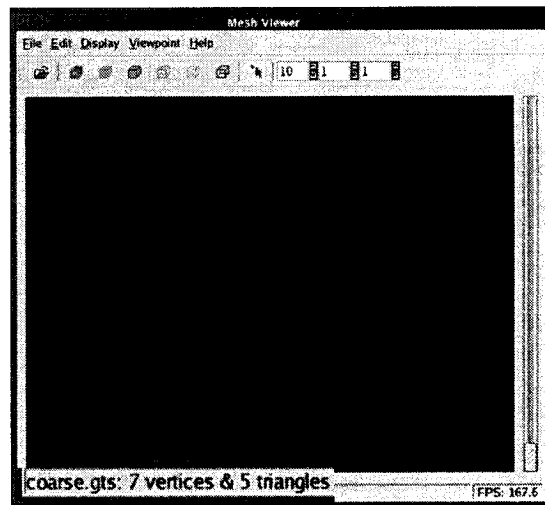


Figure 5.5. Maillage dont le triangle dégénéré a été supprimé

La plupart du temps, le fait de faire s'effondrer des facettes change considérablement la géométrie de la pièce. Par exemple, il y a un triangle dégénéré au centre de la Fig. 5.4. Les trois arêtes de ce triangle sont colinéaires et les trois angles sont  $0^\circ$ ,  $0^\circ$  et  $180^\circ$ . En détruisant le triangle central, l'algorithme détruit aussi le triangle du bas et le résultat est représenté à la Fig. 5.5. Bien que dans cet exemple le contour de la géométrie demeure le

même, sur des modèles 3D, la géométrie peut changer considérablement si le nombre de triangles à réparer est grand.

Cette solution a connus un succès limité car le fait de détruire les triangles de cette façon résulte parfois en des solides qui s'auto-intersectent et peut aussi générer des triangles dont l'orientation est renversée. De plus, parfois le fait de détruire des triangles résulte en d'autres triangles dégénérés.

Une analyse plus approfondie de ce problème a révélé que ce genre de problème fait encore l'objet de recherche dans le domaine des éléments finis. D'ailleurs, les ateliers de maillage surfacique dans Catia V5 génèrent un maillage d'une qualité relativement bonne, mais des outils permettent à l'utilisateur de réparer ce genre d'erreurs de maillage en déplaçant manuellement les nœuds.

## **5.2 Résultats obtenus lors de l'interprétation des tolérances**

L'application en réalité virtuelle ne représente qu'une ébauche d'application et ne pourrait pas être utilisée telle quelle dans un environnement de production. Cela a seulement permis de démontrer la faisabilité du concept de la génération des volumes d'existence. Un des problèmes est qu'il était nécessaire de mettre au point un module pour faire l'interprétation des tolérances. La méthode développée pour l'interprétation des tolérances devait être testée mais il n'était pas pratique d'effectuer les tests en utilisant l'application de réalité virtuelle en mode de simulation. Seul le modèle topologique est nécessaire pour l'interprétation des tolérances. La géométrie est utilisée seulement lors de la résolution. Donc, seulement le niveau topologique du modèle sera construit pour tester l'interprétation des tolérances. Ce chapitre présente une application développée afin de valider la méthode d'interprétation des tolérances. Le noyau de cet outil pourra servir de librairie pour l'interprétation du tolérancement géométrique et dimensionnel.



La méthode d'interprétation des tolérances dans un assemblage nécessite un outil pour manipuler des graphes. Il fallait soit programmer une nouvelle fonctionnalité ou récupérer une librairie «open source» sur internet. Une simple recherche dans un moteur de recherche a permis de trouver plusieurs librairies conçues dans ce but. La plupart sont basées sur le concept des templates en C++ ce qui permet de créer des graphes dont les nœuds et les arêtes peuvent être d'un type arbitraire déterminé au moment de la compilation du code. La librairie Boost Graph Librairie (BGL) a été retenue principalement pour la qualité de la documentation qui l'accompagnait. La plupart des librairies disponibles contenaient plus de fonctionnalités sur les graphes qu'il en est nécessaire pour ce projet.

Deux opérations doivent être effectuées sur les graphes : déterminer l'ensemble des chemins qui relient deux nœuds (dans le graphe d'assemblage) et déterminer les dépendances d'un nœud (dans le graphe de pièce). Le même algorithme permet de réaliser les deux opérations. Cet algorithme a été testé sur plusieurs graphes de complexités variables. Par exemple, le graphe de la Fig. 5.6 comporte des arêtes bidirectionnelles. L'objectif est d'identifier tous les chemins possibles entre le nœuds 0 et le nœud 5. L'algorithme parcourt le graphe et génère tous les chemins possibles qui débutent au nœud 0 vers n'importe quelle destination finale, pourvu que le parcours ne passe pas deux fois sur le même nœud. Le résultat est illustré à la Fig. 5.7. L'arbre au complet représenterait les dépendances du nœud 0 si c'était un graphe de pièce. Pour obtenir tous les chemins entre les nœuds 0 et 5, il suffit de filtrer ce résultat pour garder seulement les chemins qui contiennent le nœud 5.

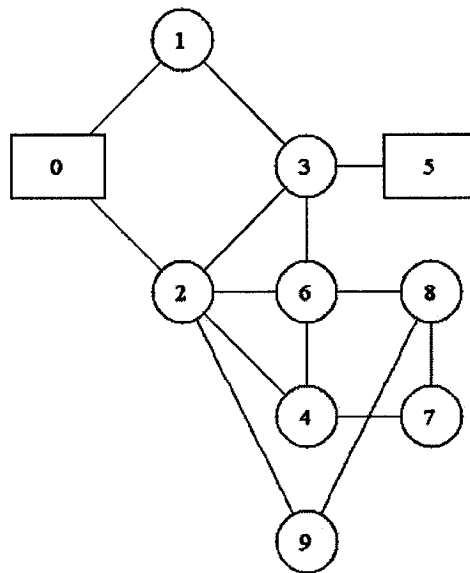


Figure 5.6. Graphe de test

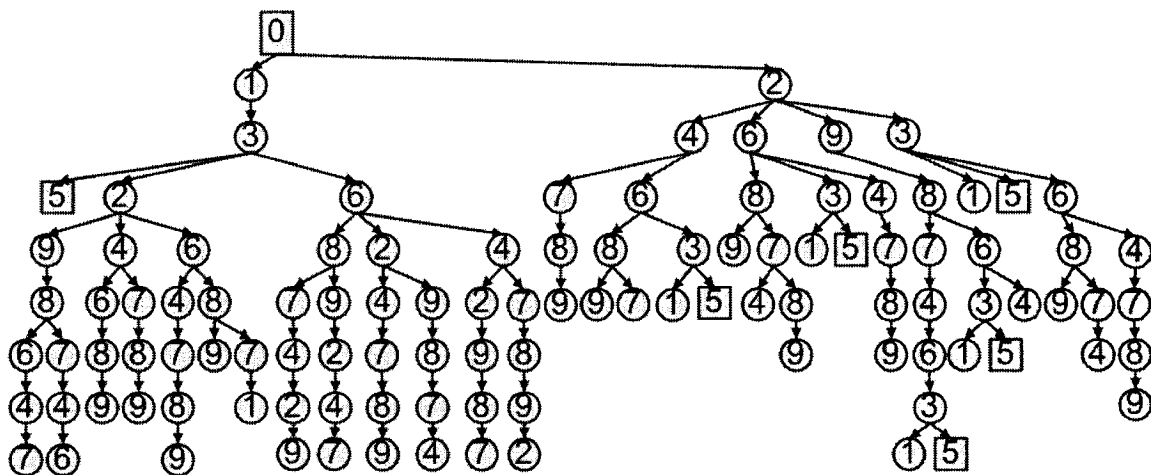


Figure 5.7. Tous les chemins possibles qui commencent au nœud 0

L'étape suivante était d'adapter les graphes afin de représenter des pièces avec tolérances et des assemblages. Puisque les classes de graphe et les algorithmes de la BGL sont des templates, cela permet de définir les structures de données de type tolérance pour les arêtes et de type élément topologique pour les nœuds du graphe de pièce, et de type pièce pour les nœuds et type liaison d'assemblage pour les arêtes du graphe d'assemblage, tel

qu'indiqué au chapitre 3. Le détail de ces structures de données est présenté au chapitre 4.

Afin d'accélérer le développement et d'effectuer un plus grand nombre d'essais qu'il aurait été possible de faire directement dans l'environnement de réalité virtuelle, un format de fichier a été développé expressément pour sauvegarder la définition d'un assemblage et les tolérances de chaque pièce. En lisant ce fichier, l'application peut représenter les structures de données de l'assemblage. Par la suite, l'utilisateur peut définir une fonction de contrôle entre deux éléments topologiques et l'application génère un rapport contenant l'information sur chaque chemin et les dépendances de chaque élément topologique utilisé.

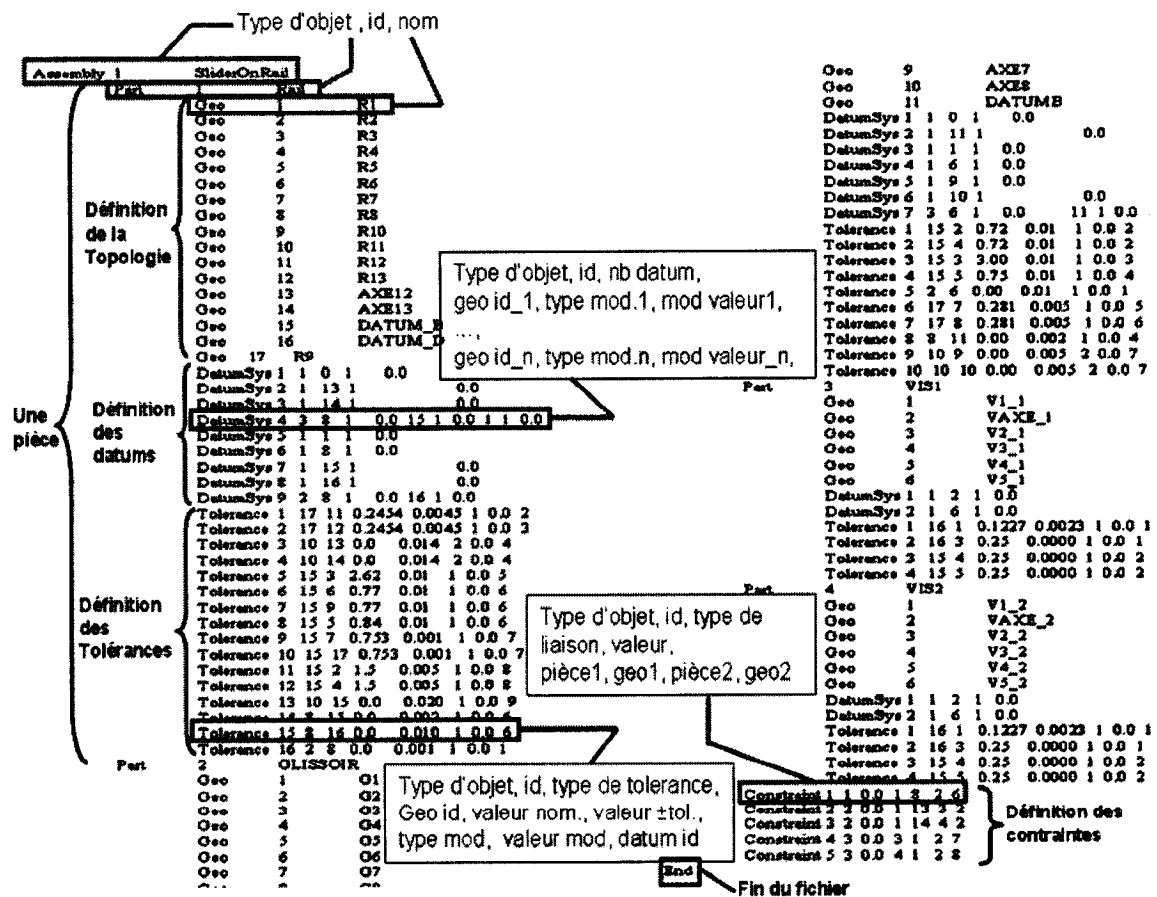


Figure 5.8. Définition du fichier de définition d'un graphe d'assemblage avec pièces tolérancées

La Fig. 5.8 indique la signification de chaque champ du fichier utilisé pour définir l'assemblage du glissoir vissé sur rail décrit à la section 3.8. La signification des codes utilisés pour représenter les différents types de la Fig. 5.8 sont présentés à l'annexe V. De plus, le fichier texte est disponible à l'annexe III. Chaque ligne du fichier commence par le type d'objet à créer et un numéro d'identification. Les objets sont créés à mesure que le fichier est lu. Par exemple, un assemblage est créé. Ensuite une pièce est ajoutée à l'assemblage et la topologie, et les tolérances sont définies pour la pièce. Une deuxième pièce est ajoutée à l'assemblage et ainsi de suite. Les éléments topologiques sont ajoutés à la pièce jusqu'à ce qu'une autre pièce soit définie ou qu'il y ait le mot clef «End» pour indiquer la fin d'un assemblage. Vers la fin du fichier, le mot clef «Constraint» est utilisé pour définir des liaisons d'assemblages.

L'utilisateur déclare ensuite une fonction de contrôle en donnant le numéro d'identification des deux pièces et des éléments topologiques qui la définissent. L'application génère alors un rapport contenant l'interprétation des tolérances pour la fonction de contrôle. La Fig. 5.9 montre les éléments principaux du rapport pour la fonction de contrôle entre l'élément R7 sur le rail et G4 sur le glissoir. Le rapport complet est disponible à l'annexe IV. Ce rapport contient les boucles cinématiques et les dépendances de chaque tolérance. On y retrouve les trois chemins décrits à la section 3.8.

Dans la Fig. 5.9, les chemins 1 et 3 ont été tronqués. Pour connaître les éléments de passage d'un chemin, il faut trouver les lignes qui indiquent seulement un élément topologique d'une pièce. Ils ont été encerclés à la Fig. 5.9, par exemple: «R7 sur rail». Ce qui suit chaque élément topologique du chemin indique l'arbre des dépendances (par exemple, voir « les dépendances de R7 » à la Fig. 5.9). Par la suite, le rapport indique soit un autre élément topologique de la même pièce qui est associé à une liaison d'assemblage, ou la liaison d'assemblage. Par exemple, à la Fig. 5.9, les dépendances de R7 sont suivies par l'élément «Axe12 on Rail» et des dépendances de l'axe 12. L'élément suivant du chemin est la liaison axe-axe qui est encadrée à la Fig. 5.9. Suivant

cette logique, la suite du rapport est l'axe de la visse 1 et la surface cylindrique de la visse 1 ainsi que leur dépendances, la liaison cylindre à cylindre, l'élément G7 du glissoir, les dépendances de G7, et finalement G4 et les dépendances de G4.

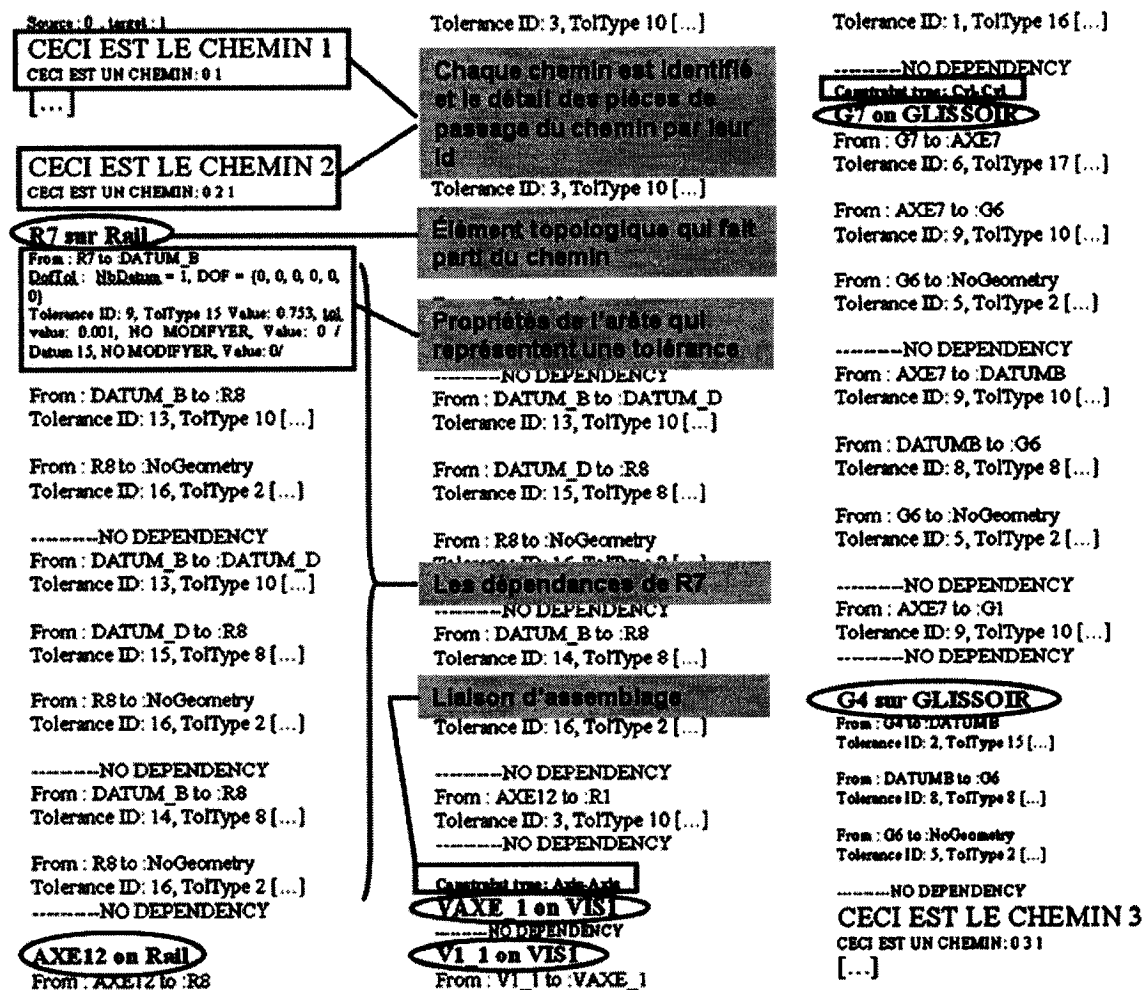


Figure 5.9. Définition du fichier de sortie

En résumé, cette application permet de vérifier la méthode d'interprétation des tolérances proposée en évitant les problèmes reliés au développement en réalité virtuelle et sans la nécessité de l'intégrer directement dans un logiciel de CAO et de modéliser les pièces complètement avant analyse. L'hypothèse est que les tolérances peuvent être extraites du logiciel et ensuite soumises à l'application dans une forme similaire à ce qui est dans le fichier d'entrée décrit. Le lien vers la géométrie qui est nécessaire pour la résolution doit

aussi être accessible. Dans le module d'annotation 3D de Catia V5, *Functional Tolerancing and Annotations*, les annotations peuvent être associées à la géométrie. Par exemple, une tolérance de parallélisme est associée à une face et sa référence à une autre face. Pareillement, une tolérance de positionnement sur un patron de trou peut être associée à chacun des trous.

## CONCLUSION ET RECOMMANDATIONS

Par la réalisation de ce travail, une méthode d'analyse graphique par génération de volume d'existence a été proposée. Un assemblage de blocs empilés les uns sur les autres a permis de démontrer le concept en générant le volume d'existence pour la surface du dessus du dernier bloc. Toutefois, l'implémentation de cette méthode dans un environnement de réalité virtuelle a soulevé plusieurs défis. Premièrement, l'analyse n'a pas pu être approfondie à des assemblages plus complexes pour plusieurs raisons. D'un côté, il y avait des problèmes de stabilité informatique lors de la génération des volumes d'existence. Deuxièmement, il fallait obtenir une meilleure interprétation du mécanisme et des tolérances afin de générer des volumes d'existence pour des assemblages plus complexes. Ce dernier point a justifié le développement de la méthode d'interprétation des tolérances. Un module d'interprétation des tolérances a été développé afin d'obtenir automatiquement toutes les boucles cinématiques pour une fonction de contrôle, et, pour chaque élément de la boucle cinématique, la hiérarchie des tolérances qui ont un effet sur l'erreur. Les résultats obtenus par le module d'interprétation sur l'assemblage du glisseur sur rail ont été utilisés afin de résoudre une fonction de contrôle en combinaison avec une méthode de résolution 1D. Cela a permis de comparer une méthode d'analyse des tolérances classique avec l'approche d'interprétation des tolérances proposée. On laisse entrevoir qu'une méthode de résolution 3D pourrait aussi être utilisée.

Les concepts développés ouvrent un nouveau champ de recherche dans l'utilisation de la réalité virtuelle pour la conception de produits et en particulier dans la capacité à voir les effets du tolérancement sur la fonctionnalité du produit. Les outils développés ne sont qu'une ébauche d'un outil d'aide à la conception en temps réel prenant en compte la fabrication, l'assemblage et le contrôle d'un produit.

Le processus de l'analyse des tolérances qui n'a pas été couvert par cette recherche est la résolution des chaînes de cotes en 3D. Les travaux futurs devraient aller en ce sens. La prochaine étape devrait consister à intégrer un solveur afin de rendre la solution

complètement opérationnelle. Afin d'économiser sur le temps de développement, il faudrait aussi développer directement dans un logiciel de CAO commercial afin de minimiser le nombre de fonctionnalités à recréer. Par la suite il pourrait être plus intéressant de migrer la solution vers un environnement de réalité virtuelle.

Tous les grands logiciels de CAO commerciaux offrent la possibilité aux clients de personnaliser leurs produits. Ceci est un moyen de mieux répondre aux besoins des clients, contribuer au développement du produit et acquérir de nouveaux marchés. Par exemple, le logiciel Catia V5 de Dassault Systèmes peut être personnalisé par du *script* VBA ou du code C++ en utilisant les bibliothèques CAA V5. Plusieurs compagnies sont déjà des partenaires CAA avec Dassault Système dans le domaine du tolérancement géométrique et dimensionnel comme Sigmatrix (CETol 6 Sigma) et DCS inc. (3DCS). Dans le domaine de la réalité virtuelle, il y a Fakespace (concepteur du CAVE System) et aussi Immersion Corporation (créateur du Cyberglove). Les avantages de développer directement dans un logiciel de CAO sont nombreux. Cela permet de réutiliser les fonctionnalités existantes pour modéliser les pièces et les assemblages. Les annotations peuvent être créées directement avec le logiciel mais interprétées et analysées par les nouvelles fonctionnalités développées. Le problème des contraintes d'assemblage serait réglé puisque cela est une fonctionnalité de l'atelier d'assemblage. De plus, les problèmes rencontrés avec la bibliothèque GTS pour les opérateurs booléens seraient peut-être réglés aussi car les opérateurs booléens dans CATIA sont beaucoup plus précis et robustes. Finalement, cela éliminerait le besoin de travailler avec des modèles en format stl puisque les données natives du logiciel seraient utilisées.

De plus, une suggestion d'outil d'analyse à réaliser serait de représenter visuellement sur les modèles 3D les chaînes de tolérances pour la fonction de contrôle qui fait l'objet de l'analyse. Cette fonctionnalité requiert très peu de travail puisque la bibliothèque d'interprétation du tolérancement est déjà capable d'obtenir cette information. La



différence est qu'avec notre application, cette information était affichée sous forme de rapport.

Finalement, dans l'optique d'approfondir la réflexion sur l'objectif initial de cette recherche, la génération du volume d'existence d'une zone de tolérance, si les outils adéquats sont disponibles, il serait intéressant de développer une méthode pour calculer le serrage ou le jeu minimum et maximum à partir des volumes d'existence de la fonction de contrôle. Par exemple, un jeu minimum pourrait être calculé en mesurant la distance minimum entre deux volumes d'existences des zones de tolérance alors qu'un serrage serait déterminé en étudiant la pénétration d'un solide dans l'autre.

## RÉFÉRENCES

- [1] Dassault Systèmes. CATIA – 3D Functional Tolerancing and Annotations 2 (FTA). Dassault Systèmes. Consultée le 6 octobre 2007, tiré de: [http://www.3ds.com/products-solutions/plm-solutions/catia/portfolio/all-products/domain/Mechanical\\_Design/product/FTA/](http://www.3ds.com/products-solutions/plm-solutions/catia/portfolio/all-products/domain/Mechanical_Design/product/FTA/)
- [2] Shen Z, Ameta G, Shah J.J, Davidson J.K (2005) A Comparative Study Of Tolerance Analysis Methods, Journal of Computing and Information Science in Engineering, 5(3):247-256.
- [3] Lin E, Zhang H-C (2001) Theoretical Tolerance Stackup Analysis Based on Tolerance Zone Analysis, The International Journal of Advanced Manufacturing Technology, 17(4):257-262.
- [4] Meadows, J (2001) Tolerance Stack-up Analysis, James D. Meadows & Associates Inc, Hendersonville.
- [5] Kandikjan T, Shah J.J, Davidson J.K (2001) A Mechanism for Validating Dimensioning and Tolerancing Schemes in CAD Systems, Computer-Aided Design, 33(10): 721-737.
- [6] Gupta S, Turner J (1993) Variational Solid Modeling for Tolerance Analysis, IEEE Computer Graphics & Applications, 13(3):64-74.
- [7] Pierce S (2003) Geometric Tolerance Analysis Methods for Imperfect-Form Assemblies, DIMACS Workshop on Computer Aided Design and Manufacturing. October 7-9, 2003, Volume 67.

- [8] Schlatter B (1996) Computer-aided 3D tolerance analysis of disk drives, IBM Journal of Research and Development, 40(5): 537-542.
- [9] Shen Z (2003) Tolerance Analysis with EDS/VisVSA, Journal of Computing and Information Science in Engineering, 3(1):95-99.
- [10] Laperrière L, Desrochers A (2001) Modeling Assembly Quality Requirements Using Jacobian or Screw Transforms: A Comparison. Proceedings of the 4<sup>th</sup> IEEE International Symposium on Assembly and Task Planning, 330-336.
- [11] Desrochers A (1991) Modèle conceptuel du dimensionnement et du tolérancement des mécanismes. Représentation dans les systèmes CFAO, Doctoral Thesis, Laboratoire de mécatronique de l'Institut Supérieur des Matériaux et de la Construction Mécanique, École Centrale de Paris.
- [12] Desrochers A, Laperrière L (2003) Spécification et analyse de tolérance assistée par ordinateur : Rétrospective des travaux. 3<sup>e</sup> journée d'étude interinstitutionnelle de fabrication haute performance, Bécancour, Canada, 21-22 novembre 2003
- [13] Desrochers A, Ghie W, Laperrière L (2003) Application of a Unified Jacobian – Torsor Model for Tolerance Analysis, Journal of Computing and Information Science in Engineering. 3: 2-14.
- [14] Ghie W (2004) Modèle unifié Jacobien-Torseur pour le tolérancement assisté par ordinateur, Thèse (Ph.D.), Université de Sherbrooke.
- [15] ADCATS (2003) Association for the Development of Computer-Aided Tolerancing Systems, ADCATS. Consulté le 6 octobre 2007, tire de : <http://adcats.et.byu.edu/>

- [16] Requicha A. A. G., Chan S.C (1986) Representation of geometric features, tolerances and attributes in solid modelers based on constructive geometry, IEEE J. Robotics & Automation, RA-2(3):156-186.
- [17] Moroni G, Requicha A (1996) Tolerance Modeling and Application Programming Interfaces, In Proc. Workshop on Concurrent Engineering (Budapest, Hungary), 13-17.
- [18] B.K.A. Ngoi, M. Agarwal, C.S. Chua. The Noded Graph Approach to Stack Analysis. The International Journal of Advanced Manufacturing Technology. 1998, 14: 343-349.
- [19] Shah J, Yan Y, Zhang B-C (1998) Dimension and Tolerance Modeling and Transformations in Feature Based Design and Manufacturing, The Journal of Intelligent Manufacturing Technology, 9: 475-488.
- [20] Kandikjan T, Shah J, Davidson J (2001) A Mechanism for Validating Dimensioning and Tolerancing Schemes in CAD Systems, Computer-Aided Design, 33(10): 721-737.
- [21] Hu J, Xiong G, Wu Z (2004) A variational geometric constraints network for a tolerance types specification, The International Journal of Advanced Manufacturing Technology, 24(3-4):214-222.
- [22] Hu J, Xiong G (2005) Dimensional and geometric tolerance design based on constraints, The International Journal of Advanced Manufacturing Technology, 26(9): 1099-1108.
- [23] Pape D (2001) The CAVE Virtual Reality System, Electronic Visualization Lab. Consulté le 6 octobre 2007, tiré de: <http://www.evl.uic.edu/pape/CAVE/>

- [24] Infiscape (2005) Vr Juggler, Infiscape. Consulté le 6 octobre 2007, tiré de : <http://www.vrjuggler.org/>
- [25] OSG Community (2007) OpenSceneGraph, AI2, Universidad Politecnica de Valencia, Spain. Consulté le 6 octobre 2007. tire de : <http://www.openscenegraph.org/projects/osg>
- [26] Stéphane Popinet (2007) The GTS Library. SourceForge.Net. Consulté le 6 octobre 2007, tire de : <http://gts.sourceforge.net/>
- [27] Merkley K.G (1998) Tolerance Analysis of Compliant Assemblies, Doctoral Thesis, Department of Mechanical Engineering, Brigham Young University.
- [28] Chase K.W (1999) Tolerance Analysis of 2-D and 3-D Assemblies, ADCATS Report No. 99-4, Consulté le 15 juin 2004, tiré de: <http://adcats.et.byu.edu/WWW/Publication/99-4/MultiDimTolAssem.pdf>.
- [29] The American Society of Mechanical Engineers (1994) Dimensioning and Tolerancing. ASME Y14.5M-1994, An American National Standard Engineering Drawing and Related Documentation Practices.
- [30] The American Society of Mechanical Engineers (1994) Mathematical Definition of Dimensioning and Tolerancing Principles, ASME Y14.5.1M-1994, An American National Standard Engineering Drawing and Related Documentation Practices, New York.
- [31] Monir M.K, Ibrahiem M.E, Rajit G, Shiv P, Khaled. A.M (2004) Geometric Dimension and Tolerance Modelling and Validation System Based On Object Oriented

Paradigm for 3D Solid Model. 2004 ASME International Mechanical Engineering Congress.

[32] Rivest L, Fortin C, Morel C (1993) Tolerancing a Solid Model with a Kinematic Formulation, *Computer-Aided Design*, 26(6):465-476.












[33] Rivest L (1993) Modélisation et analyse tridimensionnelles des tolérances dimensionnelles et géométriques. Doctoral Thesis, Université de Montréal.

[34] Foster L.W (1993) *Geo-Metrics III: The Metric Application of Geometric Dimensioning and Tolerancing Techniques*, Addison-Wesley Publishing Company Inc, Reading, Massachusetts.

[35] Wang H, Ning R, Yan Y (2005) Simulated Tolerance CAD Geometrical Model and Automatic Generation of 3D Dimension Chains, *The International Journal of Advanced Manufacturing Technology*, DOI 10.1007/s00170-005-2617-x

[36] Siek J, Lee L, Lumsdaine A (2001) *The Boost Graph Library: User Guide and Reference Manual*, Addison-Wesley Professional, Boston.

**ANNEXE I - RÈGLES DE CONCEPTION DES TOLÉRANCES  
INDÉPENDANTES POUR SEPT TYPES DE CONTRAINTES  
GÉOMÉTRIQUES VARIATIONNELLES INDÉPENDANTES  
(SVGC) [22].**























Règle	Élément de référence	Élément contraint	Contraintes géométriques variationnelles (VGCs)			Type de tolérance
			Type	$\theta$	D	
I-1	PT	Sphérique	SVGC <sub>1</sub>	$[0,0,0]^T$	$[u_i, v_i, \omega_i]^T$	$\pm$ et/ou 
I-2	SL	Cylindrique	SVGC <sub>2</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, v_i, \omega_i]^T$	$\pm$ et/ou  ou 
I-3	PL	Planaire	SVGC <sub>3</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, \omega_i]^T$	 ou $-$
I-4	PT, SL	Hélicoïdal	SVGC <sub>4</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, v_i, \omega_i]^T$	Réf. ANSI Y 14.5M
I-5	PT, SL	Révolution	SVGC <sub>5</sub>	$[0, \beta_i, \gamma_i]^T$	$[u_i, v_i, \omega_i]^T$	 ou  ou 
I-6	SL, PT	Prismatique	SVGC <sub>6</sub>	$[\alpha_i, \beta_i, \gamma_i]^T$	$[0, v_i, \omega_i]^T$	$-$ ou  ou 
I-7	PT, SL, PL	Complexe	SVGC <sub>7</sub>	$[\alpha_i, \beta_i, \gamma_i]^T$	$[u_i, v_i, \omega_i]^T$	 ou 

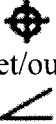
PT = point, SL = ligne, PL = plan.

**ANNEXE II - RÈGLES DE CONCEPTION DES TOLÉRANCES  
DÉPENDANTES POUR 27 TYPES DE CONTRAINTES  
GÉOMÉTRIQUES VARIATIONNELLES DÉPENDANTES (CVGC)  
[22]**

Règle	Élément de référence	Élément contrainst	Relation géométrique	Contraintes géométriques variationnelles (VGCs)			Type de tolérance
				Type	$\theta$	D	
II-1	PT	PT	coïncident	CVGC <sub>1</sub>	$[0,0,0]^T$	$[u_i, v_i, \omega_i]^T$	
II-2	PT	PT	séparé	CVGC <sub>2</sub>	$[0,0,0]^T$	$[u_i, v_i, \omega_i]^T$	$\pm$ et/ou 
II-3	PT	SL	combiné	CVGC <sub>3</sub>	$[0,0,0]^T$	$[0, v_i, \omega_i]^T$	
II-4	PT	SL	séparé	CVGC <sub>4</sub>	$[0,0,0]^T$	$[0, v_i, \omega_i]^T$	$\pm$ et/ou 
II-5	PT	PL	combiné	CVGC <sub>5</sub>	$[0,0,0]^T$	$[0,0, \omega_i]^T$	
II-6	PT	PL	séparé	CVGC <sub>6</sub>	$[0,0,0]^T$	$[0,0, \omega_i]^T$	$\pm$ et/ou 
II-7	SL	PT	combiné	CVGC <sub>7</sub>	$[0,0,0]^T$	$[0, v_i, \omega_i]^T$	
II-8	SL	PT	séparé	CVGC <sub>8</sub>	$[0,0,0]^T$	$[0, v_i, \omega_i]^T$	$\pm$ et/ou 
II-9	SL	SL	coïncident	CVGC <sub>9</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, v_i, \omega_i]^T$	 ou 
II-10	SL	SL	parallèle	CVGC <sub>10</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, v_i, \omega_i]^T$	 ou 
II-11	SL	SL	perpendiculaire	CVGC <sub>11</sub>	$[0, \beta_i, 0]^T$	$[0, v_i, 0]^T$	



II-12	SL	SL	sécante	CVGC <sub>12</sub>	$[0, \beta_i, 0]^T$	$[0, v_i, 0]^T$	 et/ou 
II-13	SL	SL	autre	CVGC <sub>13</sub>	$[0, \beta_i, 0]^T$	$[0, v_i, 0]^T$	 et 
II-14	SL	PL	combiné	CVGC <sub>14</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, \omega_i]^T$	
II-15	SL	PL	parallèle	CVGC <sub>15</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, \omega_i]^T$	 //
II-16	SL	PL	perpendiculaire	CVGC <sub>16</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, 0]^T$	 ou  ou 
II-17	SL	PL	sécante	CVGC <sub>17</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, 0]^T$	 et/ou 
II-18	PL	PT	combiné	CVGC <sub>18</sub>	$[0, 0, 0]^T$	$[0, 0, \omega_i]^T$	
II-19	PL	PT	séparé	CVGC <sub>19</sub>	$[0, 0, 0]^T$	$[0, 0, \omega_i]^T$	 et 
II-20	PL	SL	combiné	CVGC <sub>20</sub>	$[0, \beta_i, 0]^T$	$[0, 0, \omega_i]^T$	
II-21	PL	SL	parallèle	CVGC <sub>21</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, v_i, \omega_i]^T$	 et //
II-22	PL	SL	perpendiculaire	CVGC <sub>22</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, 0, 0]^T$	
II-23	PL	SL	sécante	CVGC <sub>23</sub>	$[0, \beta_i, \gamma_i]^T$	$[0, 0, 0]^T$	 et/ou 
II-24	PL	PL	coïncident	CVGC <sub>24</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, \omega_i]^T$	
II-25	PL	PL	parallèle	CVGC <sub>25</sub>	$[\alpha_i, \beta_i, 0]^T$	$[0, 0, \omega_i]^T$	 et //
II-26	PL	PL	perpendiculaire	CVGC <sub>26</sub>	$[0, \beta_i, 0]^T$	$[0, 0, 0]^T$	

II-27	PL	PL	sécante	CVGC <sub>27</sub>	$[0, \beta_i, 0]^T$	$[0, 0, 0]^T$	 et/ou
-------	----	----	---------	--------------------	---------------------	---------------	--

PT = point, SL = ligne, PL = plan.

### ANNEXE III - FICHIER D'ENTRÉE POUR L'ASSEMBLAGE DU GLISSOIR VISSÉ SUR RAIL

Voici le contenu du fichier d'entrée utilisé pour tester le module d'interprétation des tolérances. La syntaxe de ce fichier est décrite au chapitre 5 par la Fig. 5.8.

```

Assembly 1      SliderOnRail
Part      1      Rail
Geo       1      R1
Geo       2      R2
Geo       3      R3
Geo       4      R4
Geo       5      R5
Geo       6      R6
Geo       7      R7
Geo       8      R8
Geo       9      R10
Geo      10      R11
Geo      11      R12
Geo      12      R13
Geo      13      AXE12
Geo      14      AXE13
Geo      15      DATUM_B
Geo      16      DATUM_D
Geo      17      R9
DatumSys  1  1  0  1      0.0
DatumSys  2  1 13  1      0.0
DatumSys  3  1 14  1      0.0
DatumSys  4  3  8  1      0.0 15 1 0.0 1 1 0.0
DatumSys  5  1  1  1      0.0
DatumSys  6  1  8  1      0.0
DatumSys  7  1 15  1      0.0
DatumSys  8  1 16  1      0.0
DatumSys  9  2  8  1      0.0 16 1 0.0
Tolerance 1 17 11 0.2454 0.0045 1 0.0 2
Tolerance 2 17 12 0.2454 0.0045 1 0.0 3
Tolerance 3 10 13 0.0 0.014 2 0.0 4
Tolerance 4 10 14 0.0 0.014 2 0.0 4
Tolerance 5 15 3 2.62 0.01 1 0.0 5
Tolerance 6 15 6 0.77 0.01 1 0.0 6
Tolerance 7 15 9 0.77 0.01 1 0.0 6

```

Tolerance	8	15	5	0.84	0.01	1	0.0	6
Tolerance	9	15	7	0.753	0.001	1	0.0	7
Tolerance	10	15	17	0.753	0.001	1	0.0	7
Tolerance	11	15	2	1.5	0.005	1	0.0	8
Tolerance	12	15	4	1.5	0.005	1	0.0	8
Tolerance	13	10	15	0.0	0.020	1	0.0	9
Tolerance	14	8	15	0.0	0.002	1	0.0	6
Tolerance	15	8	16	0.0	0.010	1	0.0	6
Tolerance	16	2	8	0.0	0.001	1	0.0	1
Part	2	GLISSOIR						
Geo	1	G1						
Geo	2	G2						
Geo	3	G3						
Geo	4	G4						
Geo	5	G5						
Geo	6	G6						
Geo	7	G7						
Geo	8	G8						
Geo	9	AXE7						
Geo	10	AXE8						
Geo	11	DATUMB						
DatumSys	1	1	0	1	0.0			
DatumSys	2	1	11	1	0.0			
DatumSys	3	1	1	1	0.0			
DatumSys	4	1	6	1	0.0			
DatumSys	5	1	9	1	0.0			
DatumSys	6	1	10	1	0.0			
DatumSys	7	3	6	1	0.0	11	1	0.0 1 1 0.0
Tolerance	1	15	2	0.72	0.01	1	0.0	2
Tolerance	2	15	4	0.72	0.01	1	0.0	2
Tolerance	3	15	3	3.00	0.01	1	0.0	3
Tolerance	4	15	5	0.75	0.01	1	0.0	4
Tolerance	5	2	6	0.00	0.01	1	0.0	1
Tolerance	6	17	7	0.281	0.005	1	0.0	5
Tolerance	7	17	8	0.281	0.005	1	0.0	6
Tolerance	8	8	11	0.00	0.002	1	0.0	4
Tolerance	9	10	9	0.00	0.005	2	0.0	7
Tolerance	10	10	10	0.00	0.005	2	0.0	7
Part	3	VIS1						
Geo	1	V1_1						
Geo	2	VAXE_1						
Geo	3	V2_1						
Geo	4	V3_1						
Geo	5	V4_1						

```

Geo 6 V5_1
DatumSys 1 1 2 1 0.0
DatumSys 2 1 6 1 0.0
Tolerance 1 16 1 0.1227 0.0023 1 0.0 1
Tolerance 2 16 3 0.25 0.0000 1 0.0 1
Tolerance 3 15 4 0.25 0.0000 1 0.0 2
Tolerance 4 15 5 0.25 0.0000 1 0.0 2
Part 4 VIS2
Geo 1 V1_2
Geo 2 VAXE_2
Geo 3 V2_2
Geo 4 V3_2
Geo 5 V4_2
Geo 6 V5_2
DatumSys 1 1 2 1 0.0
DatumSys 2 1 6 1 0.0
Tolerance 1 16 1 0.1227 0.0023 1 0.0 1
Tolerance 2 16 3 0.25 0.0000 1 0.0 1
Tolerance 3 15 4 0.25 0.0000 1 0.0 2
Tolerance 4 15 5 0.25 0.0000 1 0.0 2
Constraint 1 1 0.0 1 8 2 6
Constraint 2 2 0.0 1 13 3 2
Constraint 3 2 0.0 1 14 4 2
Constraint 4 3 0.0 3 1 2 7
Constraint 5 3 0.0 4 1 2 8
End

```

## ANNEXE IV – RAPPORT D'INTERPRÉTATION DES TOLÉRANCES POUR L'ASSEMBLAGE DU GLISSOIR VISSÉ SUR RAIL

Voici le rapport obtenu par le module d'interprétation des tolérances pour la fonction de contrôle définie entre les faces R7 du rail et G6 du glissoir (voir l'identification des faces à la Fig. 3.17). La syntaxe de ce rapport est expliquée au chapitre 5 dans la Fig. 5.9.

Source : 0 , target : 1

CECI EST LE CHEMIN 1

CECI EST UN CHEMIN: 0 1

R7 sur Rail

From : R7 to :DATUM\_B

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 9, TolType 15 Value: 0.753, tol value: 0.001, NO MODIFIER, Value: 0 /

Datum 15, NO MODIFIER, Value: 0/

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFIER, Value: 0 /

Datum 8, NO MODIFIER, Value: 0/ / Datum 16, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFIER, Value: 0 /

Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :DATUM\_D

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : DATUM\_D to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 15, TolType 8 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /

Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 14, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /

Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

R8 on Rail

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

Constraint type: Plane-Plane

G6 on GLISSOIR

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 / Datum  
0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

G4 sur GLISSOIR

From : G4 to :DATUMB

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 2, TolType 15 Value: 0.72, tol value: 0.01, NO MODIFYER, Value: 0 /  
Datum 11, NO MODIFYER, Value: 0/

From : DATUMB to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 8, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /  
Datum 6, NO MODIFYER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}



Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 / Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

CECI EST LE CHEMIN 2

CECI EST UN CHEMIN: 0 2 1

R7 sur Rail

From : R7 to :DATUM\_B

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 9, TolType 15 Value: 0.753, tol value: 0.001, NO MODIFYER, Value: 0 / Datum 15, NO MODIFYER, Value: 0/

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 / Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 / Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :DATUM\_D

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 / Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : DATUM\_D to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 15, TolType 8 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /

Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 14, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /

Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

AXE12 on Rail

From : AXE12 to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 3, TolType 10 Value: 0, tol value: 0.014, MMC, Value: 0 / Datum 8, NO

MODIFYER, Value: 0/ / Datum 15, NO MODIFYER, Value: 0/ / Datum 1, NO

MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : AXE12 to :DATUM\_B

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 3, TolType 10 Value: 0, tol value: 0.014, MMC, Value: 0 / Datum 8, NO  
MODIFYER, Value: 0/ / Datum 15, NO MODIFYER, Value: 0/ / Datum 1, NO  
MODIFYER, Value: 0/

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :DATUM\_D

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : DATUM\_D to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 15, TolType 8 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 14, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : AXE12 to :R1

DofTol : NbDatum = 3, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 3, TolType 10 Value: 0, tol value: 0.014, MMC, Value: 0 / Datum 8, NO  
MODIFYER, Value: 0/ / Datum 15, NO MODIFYER, Value: 0/ / Datum 1, NO  
MODIFYER, Value: 0/

-----NO DEPENDENCY

Constraint type: Axis-Axis

VAXE\_1 on VIS1

-----NO DEPENDENCY

V1\_1 on VIS1

From : V1\_1 to :VAXE\_1

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 1, TolType 16 Value: 0.1227, tol value: 0.0023, NO MODIFYER, Value:  
0 / Datum 2, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

Constraint type: Cyl-Cyl

G7 on GLISSOIR

From : G7 to :AXE7

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 6, TolType 17 Value: 0.281, tol value: 0.005, NO MODIFYER, Value: 0 /  
Datum 9, NO MODIFYER, Value: 0/

From : AXE7 to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 9, TolType 10 Value: 0, tol value: 0.005, MMC, Value: 0 / Datum 6, NO  
MODIFYER, Value: 0/ / Datum 11, NO MODIFYER, Value: 0/ / Datum 1, NO  
MODIFYER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 / Datum  
0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : AXE7 to :DATUMB

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 9, TolType 10 Value: 0, tol value: 0.005, MMC, Value: 0 / Datum 6, NO  
MODIFYER, Value: 0/ / Datum 11, NO MODIFYER, Value: 0/ / Datum 1, NO  
MODIFYER, Value: 0/

From : DATUMB to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 8, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /  
Datum 6, NO MODIFYER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 / Datum  
0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : AXE7 to :G1

DofTol : NbDatum = 3, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 9, TolType 10 Value: 0, tol value: 0.005, MMC, Value: 0 / Datum 6, NO  
MODIFYER, Value: 0/ / Datum 11, NO MODIFYER, Value: 0/ / Datum 1, NO  
MODIFYER, Value: 0/

-----NO DEPENDENCY

G4 sur GLISSOIR

From : G4 to :DATUMB

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 2, TolType 15 Value: 0.72, tol value: 0.01, NO MODIFYER, Value: 0 /  
Datum 11, NO MODIFYER, Value: 0/

From : DATUMB to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 8, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /  
Datum 6, NO MODIFYER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 / Datum  
0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

CECI EST LE CHEMIN 3

CECI EST UN CHEMIN: 0 3 1

R7 sur Rail

From : R7 to :DATUM\_B

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 9, TolType 15 Value: 0.753, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 15, NO MODIFYER, Value: 0/

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :DATUM\_D

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : DATUM\_D to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 15, TolType 8 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /  
Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 14, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /  
Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}



Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFIER, Value: 0 /  
Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

AXE13 on Rail

From : AXE13 to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 4, TolType 10 Value: 0, tol value: 0.014, MMC, Value: 0 / Datum 8, NO  
MODIFIER, Value: 0/ / Datum 15, NO MODIFIER, Value: 0/ / Datum 1, NO  
MODIFIER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFIER, Value: 0 /  
Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

From : AXE13 to :DATUM\_B

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 4, TolType 10 Value: 0, tol value: 0.014, MMC, Value: 0 / Datum 8, NO  
MODIFIER, Value: 0/ / Datum 15, NO MODIFIER, Value: 0/ / Datum 1, NO  
MODIFIER, Value: 0/

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFIER, Value: 0 /  
Datum 8, NO MODIFIER, Value: 0/ / Datum 16, NO MODIFIER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /

Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :DATUM\_D

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 13, TolType 10 Value: 0, tol value: 0.02, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/ / Datum 16, NO MODIFYER, Value: 0/

From : DATUM\_D to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 15, TolType 8 Value: 0, tol value: 0.01, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFYER, Value: 0 /

Datum 0, NO MODIFYER, Value: 0/

-----NO DEPENDENCY

From : DATUM\_B to :R8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 14, TolType 8 Value: 0, tol value: 0.002, NO MODIFYER, Value: 0 /

Datum 8, NO MODIFYER, Value: 0/

From : R8 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 16, TolType 2 Value: 0, tol value: 0.001, NO MODIFIER, Value: 0 /  
Datum 0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

From : AXE13 to :R1

DofTol : NbDatum = 3, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 4, TolType 10 Value: 0, tol value: 0.014, MMC, Value: 0 / Datum 8, NO  
MODIFIER, Value: 0/ / Datum 15, NO MODIFIER, Value: 0/ / Datum 1, NO  
MODIFIER, Value: 0/

-----NO DEPENDENCY

Constraint type: Axis-Axis

VAXE\_2 on VIS2

-----NO DEPENDENCY

V1\_2 on VIS2

From : V1\_2 to :VAXE\_2

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 1, TolType 16 Value: 0.1227, tol value: 0.0023, NO MODIFIER, Value:  
0 / Datum 2, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

Constraint type: Cyl-Cyl

G8 on GLISSOIR

From : G8 to :AXE8

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 7, TolType 17 Value: 0.281, tol value: 0.005, NO MODIFIER, Value: 0 /  
Datum 10, NO MODIFIER, Value: 0/

From : AXE8 to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 10, TolType 10 Value: 0, tol value: 0.005, MMC, Value: 0 / Datum 6, NO  
MODIFIER, Value: 0/ / Datum 11, NO MODIFIER, Value: 0/ / Datum 1, NO  
MODIFIER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFIER, Value: 0 / Datum  
0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

From : AXE8 to :DATUMB

DofTol : NbDatum = 2, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 10, TolType 10 Value: 0, tol value: 0.005, MMC, Value: 0 / Datum 6, NO  
MODIFIER, Value: 0/ / Datum 11, NO MODIFIER, Value: 0/ / Datum 1, NO  
MODIFIER, Value: 0/

From : DATUMB to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 8, TolType 8 Value: 0, tol value: 0.002, NO MODIFIER, Value: 0 /  
Datum 6, NO MODIFIER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFIER, Value: 0 / Datum  
0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

From : AXE8 to :G1

DofTol : NbDatum = 3, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 10, TolType 10 Value: 0, tol value: 0.005, MMC, Value: 0 / Datum 6, NO  
MODIFIER, Value: 0/ / Datum 11, NO MODIFIER, Value: 0/ / Datum 1, NO  
MODIFIER, Value: 0/

-----NO DEPENDENCY

G4 sur GLISSOIR

From : G4 to :DATUMB

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 2, TolType 15 Value: 0.72, tol value: 0.01, NO MODIFIER, Value: 0 /  
Datum 11, NO MODIFIER, Value: 0/

From : DATUMB to :G6

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 8, TolType 8 Value: 0, tol value: 0.002, NO MODIFIER, Value: 0 /  
Datum 6, NO MODIFIER, Value: 0/

From : G6 to :NoGeometry

DofTol : NbDatum = 1, DOF = {0, 0, 0, 0, 0, 0}

Tolerance ID: 5, TolType 2 Value: 0, tol value: 0.01, NO MODIFIER, Value: 0 / Datum  
0, NO MODIFIER, Value: 0/

-----NO DEPENDENCY

## ANNEXE V - SIGNIFICATION DES CODES NUMÉRIQUES UTILISÉS DANS LE FICHIER D'ASSEMBLAGE

Type de tolérance		Type de modificateur		Type de liaison d'assemblage	
Rectitude	1	Aucun	1	Plan à plan	1
Planéité	2	MMC	2	Axe à axe	2
Circularité	3	LMC	3	Cylindre à cylindre	3
Cylindricité	4	Projection	4		
Profil d'une ligne	5	État libre	5		
Profil d'une surface	6	Plan tangent	6		
Inclinaison	7	Diamètre	7		
Perpendicularité	8	Diamètre d'une sphère	8		
Parallélisme	9	Rayon	9		
Position	10	Rayon d'une sphère	10		
Concentricité	11	Contrôle du rayon (CR)	11		
Symétrie	12	Référence	12		
Battement circulaire	13	Longueur d'arc	13		
Battement total	14	Tolérance statistique	14		
Distance	15	Entre	15		
Rayon	16				
Diamètre	17				